Elmi Xəbərlər Jurnalı *Nömrə 1, 2024. 6-13*
DOI:10.30546/gdu.1.2024.6

RİYAZİYYAT VƏ MEXANİKA

UOT:372.0:002

ALQORİTMLƏRİN HAZIRLANMASI METODLARI

Babayeva Şəkər Maarif qızı
shakarbabayeva@mail.ru
Gəncə Dövlət Universiteti

Xülasə: *Alqoritmlər elmi və texniki sahədə mühüm əhəmiyyət kəsb edir. Alqoritmlərin analizi və hazırlanması üsullarına həsr olunmuş kursun tədris edilməsində universitet təhsilinin əsas məqsədi tələbələrdə qoyulan məsələlərin sərbəst həlli vərdişlərinin yaradılmasıdır.*

Alqoritmlərin hazırlanması metodlarını məsələlər həllinin universal vasitəsi kimi hesab etmək olar. Onların tətbiqi hesablama texnikası və riyaziyyatın ənənəvi məsələri ilə məhdudlaşmır. Alqoritmin hazırlanması metodu – hesablama texnikasının müxtəlif sahələrinə aid geniş spektrli məsələlərin alqoritmik həlli üçün tədbiq olunan universal yanaşmadır. Bu məqalədə alqoritmlər və onların hazırlanması metodlarının öyrənilməsinin vacibliyi araşdırılır, alqoritmlərin hazırlanması üçün mövcud olan əsas metodların icmalı təqdim edilir və həmçinin onların hesablama texnikasının klassik məsələlərinin həlli üçün tətbiqinə dair bəzi nümunələr təqdim edilir.

Açar sözlər: alqoritmlərin hazırlanması metodları, kobud güc metodu, hərtərəfli axtarış, dekompozisiya metodu, məsələnin ölçüsünün azaldılması metodu, dinamik proqramlaşdırma, xəsis aıqoritmlər, qayıdışla axtarış metodu.

Ключевые слова: методы разработки алгоритмов, метод грубой силы, исчерпывающий перебор, метод декомпозиции, метод уменьшения размера задачи, динамическое программирование, жадные алгоритмы, метод поиска с возвратом.

Key words: methods for constructing algorithms, brute force method, exhaustive search, decomposition method, problem's size reduction method, dynamic programming, greedy algorithms, backtracking method

Alqoritmlər elmi və texniki sahədə mühüm əhəmiyyət kəsb edir. Ona görə alqoritmlər və onların analizinə aid çoxlu sayda ədəbiyyat işlənib hazırlanmışdır. Belə kitabları iki böyük qrupa ayırmaq olar. Bunlardan birincisində alqoritmər həll olunan məsələnin tipinə görə təsnif olunur. Bir qayda olaraq belə kitablarda nizamlama, axtarış, qrafların emal edilməsi alqoritmləri, kombinator alqoritmlər və s. tiplərinə ayrıca fəsillər həsr olunur. Belə yanaşmanın üstünlüyü ondan ibarətdir ki, o, birbaşa məsələlərin həlli üçün müxtəlif alqoritmlərin effektivliyini qiymətləndirməyə imkan verir. Çatışmazlıq cəhəti isə ondadır ki, belə yanaşmada diqqət alqoritmlərin qurulması metodoloqiyasina yox, məsələnin həll edilməsinə yönəldilir.

Ikinci, alternativ yanaşmanın istifadəsində əsas diqqət alqoritmlərin hazırlanması metodikasına yetirilir. Belə kitablarda hesablama texnikasının müxtəlif sahələrinə aid olan alqoritmlər onların hazırlanmasında eyni metodun istifadəsinə görə qruplaşdırılır. Kitabın belə təşkili alqoritmlərin hazırlanması və analizinə həsr olunmuş əsas kursa daha çox məqsədəuyğun olur. Alqoritmlərin hazırlanması metodologiyasına daha çox diqqət yetirilməsinin bir neçə səbəbi var. Birincisi, tələbələr bunu tanış olmayan məsələnin həlli üçün alqoritmlərin hazırlanmasında tətbiq edə bilərlər. İkincisi, onlar bütün tanış olduqları alqoritmləri hazırlanması ideyasına uyğun təsnif etməyi bacarmalıdır. İnformatika sahəsi üzrə təhsilin əsas məqsədi müxtəlif tətbiq sahələrinə aid olan alqoritmlərin hazırlanmasının

DOI:10.30546/gdu.1.2024

ümumi ideyalarının oyrənilməsi olmalıdır. Nəhayət, ixtiyarı elmi fənnin öyrənilməsində əsas diqqət əsas anlayışlar siteminə yönəldilməlidir. Üçüncüsü, alqoritmlərin hazırlanması metodologiyasının öyrənilməsi mühüm əhəmiyyət kəsb edir, çünki o, informatika sahəsinə aid olan məsələlərin ümumi həllinin axtarışı metodikasının dərk edilməsində açar rolunu oynayır.

Alqoritmlərin hazırlanması metodlarını məsələlər həllinin universal vasitəsi kimi hesab etmək olar. Onların tətbiqi hesablama texnikası və riyaziyyatın ənənəvi məsələri ilə məhdudlaşmır. Bunun vacibliyi iki faktorla təstiq olunur. Birincisi, daha çox kompüter proqramları ənənəvi tətbiq sahəsindən kənara çıxır və ümid edilir ki, bu tendensiya davam edəcək. İkincisi, universitet təhsilinin əsas məqsədi tələbələrdə qoyulam məsələlərin sərbəst həlli vərdişlərinin yaradılmasıdır. Ona görə informatikaya aid proqramlar çərçivəsində alqoritmlərin hazırlanması və analizə həsr olunmuş kurs bunun üçün idealdır, çünki tələbələrdə xüsusi məsələ həll etmə bacarıqlarını yaradır.

Qoyulmuş məsələlərin həll alqoritmlərini necə hazırlamaq lazımdır? Bu suala cavab vermək üçün alqoritmlərin hazırlanmasının ümumi metodlarını oyrənmək məsləhətdir. Alqoritmin hazırlanması metodu nədir? Alqoritmin hazırlanması metodu (algorithm design technique) – hesablama texnikasının müxtəlif sahələrinə aid geniş spektrli məsələlərin alqoritmik həlli üçün tədbiq olunan universal yanaşmadır. Bu metodların oyrənilməsi aşağıdakı səbəblərə yüksək dərəcədə vacibdir.

Birincisi, yeni məsələlərin (yəni elə məsələrin ki, onların həlli üçün kifayət qədər yaxşı alqoritmlər mövcud deyil) həlli alqoritmlərin universal prisiplər dəstilə istiqamətlənərək hazırlanmasını təmin edir. Belə metodların oyrəniməsi (məşhur bir kəlamın təsfiri ilə) təklif olunan balığa yox, bağışlanılmış balıq tutmaq üçün qarmağa bənzərdir. Əlbəttə ki, bu universal metodların hamısı rast gələn praktik məsələlərin həlli üçün əlverişli olmayacaq. Ancaq bunlar təhsildə və işinizdə faydalı olacaq güclü alətlər toplusududur.

Ikincisi, alqoritmlər informatikanın təməl daşıdır. Əsas anlayışların təsnifatı ixtiyarı elm üçün vacibdir və informatika istisna deyil. Bu metodların öyrənilməsi alqoritmləri, əsasında olan hazırlanması prinsipinə uyğun olaraq təsnif etməyə imkan verir. Ona görə bu metodlar alqoritmlərin həm təsnifatına, həm öyrənilməsinə daha çox münasibdir.

Məsələrin həlli üçün alqoritmlərin qurulması yaradıcılıq işidir. Ixtiyarı alqoritmlərin asanlıqla qurulmasına imkan verən universal üsul yoxdur. Alqoritmləşdirmədə əsas yanaşma məsələni alt məsələlərə bölünməsidir. Bu təbiidir, çünki bir addımı elementar addımlar ardıcıllığına çevirmək lazımdır. Bu çevrilmənin özü də bir neçə mərhələdən ibarət ola bilər, burada yeganə addım bir neçə sadə, lakin hələ elementar olmayanlara bölünə bilər. Bu sadə addımlar xüsusi məsələlərə (alt məsələlərə) uyğun gəlir, onların məcmu (hamısının) həlli əsas (ilkin) məsələnin həllinə gətirib çıxarır.

Alqoritmlərin hazırlanmasının müxtəlif meotdları məsələnin hansı alt məsələlərə bölünməsi və bu alt məsələlərin bir-biri ilə əlaqəsi ilə fərqlənir. Alqoritmlərin hazırlanması metodlarının ümumi qəbul edilmiş təsnifatı olmasa da, bununla bağlı bəzi ümumi mülahizələr irəli sürmək olar.

Hal hazırda böyük sinif məsələlərin həlli üçün effektiv alqoritmlərin alınmasına imkan verən bir sıra metodlar möcuddur. Bunlardan ən mühümlərinə primitiv (kobud güç) metodu, dekompozisiya metodu, dinamik proqramlaşdırma, xəsis alqoritmlər, məsələnin ölçüsünün azaldılması, transformasiya, qayıdışla axtarış, lokal axtarış metodları və s. aiddir.

Kobud güc (brute force) metodu məsələnin həllinə bilavasitə məsələnin ifadə edilməsinə və istifadə olunan konsepsiyaların tərifləriнə əsaslanan birbaşa yanaşmadır. Strateqiyanın təyinində “ güc ” deyəndə, ağlın, zehnin gücü yox, kompüterin gücü nəzərdə tutulur, yəni “Gücün var, ağıl lazım deyil ” zərb-məsəldəki güc nəzərə alınır. Bu strateqiyanı

DOI:10.30546/gdu.1.2024

başqa sözlərlə daha sadə ifadə etmək olar: " Fikirləşmək yox, hərəkət etmək lazımdır." Bir çox hallarda güc metodunun tətbiqi daha asan olur.

Misal üçün qüvvətə yüksəltmə məsələsinə baxaq: ixtiyarı a və mənfi olmayan tam n üçün a^n -in hesablanması məsələsi. Bu məsələ adi gəlsə də alqoritmlərin hazırlanmasının bir neçə metodun , o cümlədə güc metodunun izahını verməyə imkan verir. (Qeyd edək ki, $a^n \bmod m$ qiymətinin böyük tam ədədlər üçün hesablanması aparıcı şifrləmə alqoritmlərinin əsas komponentidir.) Qüvvətə yüksəltmənin tərifinə görə :

$$a^n = \underbrace{a \cdot a \cdot \ldots \cdot a}_{n \text{ dəfə}}.$$

Buradan a^n -in hesablamasının ən sadə alqoritmini (a^n-in 1-ə bərabər olan ilkin qiymətini dövrün içində a-ya n dəfə vurulması nəticəsində) alırıq.

Güc metodunun istifadəsinə misal olaraq ən böyük ortaq bölənin ($gcd(m,n)$) tapılması üçün n - dən başlayaraq, ondan kiçik bütün tam ədədlərin yoxlanması üsulu ilə alqoritmi, matrislərin vurulmasının tərifinə əsaslanan alqoritmi, ardıcıl axtarış alqoritmi, seçim üsulu ilə nizamlama alqoritmi, qabarcıqlı nizamlama alqoritmi, alt sətirlərin axtarışının sadə alqoritmi və s. göstərmək olar.

Kobud güc metodun istifadəsi nadir hallarda effektiv alqoritm qurmağa imkan verməyinə baxmayaraq onun öyrənilməsinə diqqət yetirməmək olmaz, çünki bu metod özü ilə alqoritmlərin qurulmasının vacib strategiyasını ifadə edir. Birincisi, digər strategiyalardan fərqli olaraq güc metodu geniş diapazonlu məsələlərin həllinə tətbiq edilir.(Oxşar ki, bu yeganə yanaşmadır ki, həllinə tətdiq edilə bilməyən məsələni göstərmək daha çətin olar.) Məsələn, güc metodu bir çox elementar, ancaq vacib olan – n ədədin cəminin hesablanması, siyahıda ən böyük elementin axtarışı və bu kimi alqoritmik məsələlərin həlli üçün ustifadə edilir. İkincisi, bəzi vacib olan məsələlər üçün (məsələn, nizamlama, axtarış, matrislərin vurulması, alt sətirlərin axtarışı) güc metodunun istifadəsilə kifayət qədər rasional alqoritmlər qurulur. Üçüncüsü, əgər az sayda məsələ nüsxəsinin həlli lazım olarsa, daha effektiv alqoritmin hazırlanmasının qiyməti qəbul edilməz ola bilər, ancaq güc metodunun tətbiqi daha ucuz başa gəlir (çünki hazırlanmasına çox vaxt sərf edilmir). Dördüncüsü, ümumi halda effektiv olmayan güc metodu ölçüsü böyük olmayan məsələ nüsxələri üçün əlverişli ola bilər. Nəhayət, güc metodunun istifadəsilə qurulan alqoritm vacib nəzəri və ya didaktik məqsədlərə qulluq edə bilər, məsələn, bu alqoritmin effektivliyini verilən məsələnin həlli üçün qurulan digər alqoritmlərin effektivliyilə müqayisə etmək olar.

Kobud güc metodunun istifadəsilə yanaşmanın əsas üstünlükəri onun geniş tətbiq edilməsi və sadəliyidir, əsas catışmazlığı isə belə alqoritmlərin effektivliyinin çox az olmasıdır. Məsələnin həlli üçün güc metodunun tədbiqi adətən kifayət qədər kiçik səy bahasına təkmilləşdirilə bilən alqoritm verir.

Hərtərəfli axtarış (exhaustive search – bütün mümkün halların axtarışı (nəzərdən keçirilməsi)) kombinator məsələlərin həlli üçün güc metodunun istifadəsilə yanaşmadır. O, məsələnin kombinator obyektlərinin generasiyasını nəzərdə tutur, onların içində məsələnin şərtlərini ödəyənlərini seçir və sonra lazım olan obyekti axtarır.

Hərtərəfli axtarış yanaşmasının tətbiqi ilə həll oluna bilən məsələlərin tipik nümayandəsinə tacir məsələsi, çanta məsələsi, tapşırıqlar məsələsi aid edilir. Hərtərəfli axtarış yanaşması belə məsələlərin kiçik nüsxələrinə tətbiq edilə bilər (hamısına tətbiqi əlverişli deyil).

Dekompozisiya (parçalanma ("divide and conquer" və ya "parçala və hökm et")) metodu alqoritmlərin hazırlanmasının ən populyar metodudur. Bir sıra çox effektiv

DOI:10.30546/gdu.1.2024

alqoritmlər bu ümumi strategiyanın istifadəsilə alınmışdır. Parçalanma metoduna əsaslanan alqoritmlər aşağıda verilmiş plana uyğun işləyir:

1. Ilkin məsələ bir neçə kiçik məsələlərə bölünür (ideal halda eyni ölçülü məsələlərə).
2. Kiçik məsələlər həll olunur (adətən, rekursiyanın istifadəsilə, bəzən kiçik məsələlər üçün hansısa digər alqoritm tətbiq olunur).
3. Ilkin məsələnin həlli kiçik məsələlərin həllərinin kombinasiyası nəticəsində alınır.

Dekompozisiya metodunun tətbiqinə misal olaraq qovuşma üsulu ilə nizamlama, cəld nizamlama, binar axtarış, böyük tam ədədlərin vurulması (müasir kriptologiyada 100-dən çox rəqəmləri olan ədədləri vurmaq tələb olunur), kvadrat matrislərin vurulması üçün Ştrassen alqoritmi, hesablama həndəsənin vacib məsələləri (ən yaxın nöqtələr cütü ilə bağlı və qabarıq çərçivə ilə bağlı məsələlər) və s. göstərmək olar. Qeyd etmək lazımdır ki, dekompozisiya metodu paralel hesablamaların aparılması üçün idealdır: hər bir alt məsələ eyni vaxtda öz prosessoru ilə həll edilir.

Məsələ ölçüsünün azaldılması (“decrease and conquer” – “azalt və hökm et”) metodu verilmiş məsələnin və həmin məsələnin daha kiçik nüsxəsi arasındakı əlaqənin istifadəsinə əsaslanır. Əgər belə əlaqə təyin edilib (qurulub), onda onu ya “yuxarıdan aşağıya” (rekursiv) ya da “aşağıdan yuxarı” (rekursiyasız) istifadə etmək olar. Məsələ ölçüsünün azaldılması metodunun üç əsas variantı mövcuddur:

- sabit qiymətə azaltma;
- sabit vuruğa azaltma;
- dəyişən qiymətə azaltma.

Sabit qiymətə azaltma variantında məsələ nüsxəsinin ölçüsü alqoritmin hər iterasiyasında eyni qiymətə azalır, adətən, bir vahidə azalır. Misal üçün tam müsbət qüvvəti olan a^n -in hesablanması məsələsinə baxaq. n ölçülü məsələnin həlli ilə $n-1$ ölçülü məsələ nüsxəsinin həlli ilə əlaqəsi aşkar düsturla $a^n = a^{n-1} \cdot a$ ifadə edilir. Beləliklə, $f(n)=a^n$ funksiyası ya “yuxarıdan aşağıya” aşağıdakı rekursiv tərifdən itifadə etməklə hesablana bilər:

$$f(n)=\begin{cases} f(n-1)\cdot a, & \text{əgər } n>1 \\ a, & \text{əgər } n=1 \end{cases}$$

və ya rekursiyasız “aşağıdan yuxarı”, $n-1$ dəfə a -nı özünə vurmaqla (köbud qüc metodunun tədbiqindəki kimi) hesablanır.

Yerinə qoyma üsulu ilə nizamlama alqoritmi, qrafların eninə və dərinliyə doğru keçilməsi alqoritmləri də məsələnin ölçüsünün bir vahidə azaldılması metodunun istifadəsilə hazırlanıb.

Sabit vuruğa azaltma variantında məsələ nüsxəsinin ölçüsü alqoritmin hər bir iterasiyasında eyni vuruğa azalır. Çox vaxt bu vuruq 2-yə bərabər olur (məsələn, ikilik axtarış alqoritmində). Misal üçün yenə qüvvətə yüksəltmək məsələsinə baxaq: **n**-ölçülü məsələ nüsxəsinin həlli (yəni a^n -in hesablanması) ilkin məsələ ilə aşkar münasibətlə $a^n = (a^{n/2})^2$ əlaqəli olan yarım ölçülü məsələ həll edilir, yəni $a^{n/2}$ -in qiyməti hesablanır. Lakin qüvvəti tam müsbət ədəd olan məsələyə baxdığımıza görə bu metod ancaq cüt n-lər üçün yarıyır. Əgər n tək ədəd olarsa, onda biz a^{n-1}-i qüvvəti cüt ədəd olan qayda ilə hesablamalıyıq, sonra isə nəticəni a -ya vurmalıyıq. Beləliklə, aşağıdakı düstura əməl edilməlidir:

$$a^n = \begin{cases} (a^{n/2})^2, & \text{əgər } n \text{ cütdür} \\ (a^{(n-1)/2})^2 \cdot a, & \text{əgər } n \text{ təkdir və } n>1 \\ a, & \text{əgər } n=1 \end{cases}$$

Saxta sikkənin tapılması, natural ədədlərin rusca vurulması (Rusian peasant method), İosifin məsələsi (Josephus problem) alqoritmlərin hazırlanması da məsələnin ölçüsünün sabit vuruğa (2-yə) azaldılması metodunun tətbiqi ilə aparılır.

DOI:10.30546/gdu.1.2024

Dəyişən qiymətə azaltma variantında məsələnin ölçüsünün azalma qiyməti iterasiyadan iterasiyaya dəyişilir. Buna misal olaraq ən böyük ortaq bölənin axtarılması üçün Evklid alqoritmini qətirmək olar. Xatırladaq ki, bu alqoritm

gcd (m,n) = gcd (n,m mod n) düsturuna əsaslanır.

Tənliyin sağ tərəfindəki arqumentləri sol tərəfindəki arqumentlərindən həmişə (ikinci iterasiyadan başlayaraq) kiçik olur, lakin onlar nə sabit qiymətə, nə də sabit vuruq qiymətinə azalmır. Nizamlanmış massivdə interpolyasiya axtarışı (interpolation search) və binar axtarış ağacında axtarış alqoritmləri də bu metodunun (dəyişən qiymətə azaltma variantının) istifadəsi ilə hazırlanıb.

Transformasiya ("transform and conquer" – "çevir və hökm et") metodu iki mərhələdə yerinə yetirilir: əvvəlcə məsələ daha asan və həlli məlum olan məsələyə gətirilir (çevirilir) və sonra isə alınan məsələ həll olunur. Bu metodun çevrilmə üsulu ilə fərqlənən üç əsas variantı mövcuddur:

- məsələnin sadələşdirilməsi,
- məsələnin təsvirinin dəyişdirilməsi,
- məsələni alqoritmi məlum olan digər məsələyə gətirilməsi.

Məsələnin sadələşdirilməsi üsulunda məsələ elə xüsusi xassələri olan məsələyə çevrilir ki, məsələnin həlli sadələşir. Məsələn, bu variantın tətbiqi nümunələri kimi həllindən qabaq nizamlama, xətti tənliklər sistemini həll etmək üçün Qauss üsulu (əmsalları yoxetmə üsulu) və AVL-ağaclarını göstərmək olar.

Məsələnin təsvirinin dəyişdirilməsi üsuluna piramidavari nizamlama, çoxhədlinin hesablanması üçün Horner sxemi, qüvvətə yüksəltmənin binar alqoritmlərini misal göstərmək olar.

Transformasiya metodunun *məsələni alqoritmi məlum olan digər məsələyə gətirilməsi* variantına məsələnin xətti proqramlaşdırma məsələsinə və qraf məsələsinə gətirilməsini misal göstərmək olar.

Dinamik proqramlaşdırma – optimallaşdırmanın ən güclü metodlarından biridir. Bu metodun yaradılması amerikalı alim R.Bellmanın adı ilə bağlıdır. O, XX əsrin 50-ci illərində konkret məsələlərin həllində, sonralar optimallıq prinsipi adlandırılan priyom istifadə etdi. Bu prinsipin tətbiqinin əsas sahəsi dəyişilən çoxaddımlı proseslərdir, ona görə yeni optimallaşdırma metodu *dinamik* adlandırıldı. Dinamik olmağı ilə bu metod məsələlərinin qoyuluşu statik xarakteri olan xətti və riyazi proqramlaşdırmadan fərqlənirdi.

"Dinamik proqramlaşdırma" söz birləşməsində olan "proqramlaşdırma" sözünün "ənənəvi" proqramlaşdırmaya (kodun yazılmasına) aidiyyəti yoxdur və o, "optimallaşma" sözünün sinonimi olan "riyazi proqramlaşdırma" söz birləşməsində olan mənanı daşıyır. Ona görə "proqram" sözü bu mövzuda məsələ həllinin alınması üçün əməliyyatların optimal ardıcıllığı kimi başa düşülür.

Bellmanın sözlərinə görə, üsulun əsas çatışmazlığı ondan ibarətdir ki, məsələnin ölçüsü artdıqca üsulun mürəkkəbliyi həddindən çox artır. Dinamik proqramlaşdırma vasitəsi ilə eksponensial mürəkkəbliyə malik olan alqoritmləri polinomial mürəkkəbliyə malik olan alqoritmlərə gətirmək olar.

Dinamik proqramlaşdırma üst-üstə düşən alt məsələlərlə bağlı məsələlərin həlli metodudur. Adətən, bu cür alt məsələlər, qoyulan məsələnin həllini eyni tipli daha kiçik məsələlərin həlli ilə birləşdirən rekursiv münasibətlərdən yaranır. Dinamik proqramlaşdırma kiçik alt məsələlərin hər birinin həllinin yalnız bir dəfə tapılmasını və ***nəticələrin cədvələ yazılmasını*** nəzərdə tutur, daha sonra ilkin məsələnin həlli buradan alınır.

DOI:10.30546/gdu.1.2024

Dinamik proqramlaşdırmanın optimallaşdırma məsələlərinə tətbiqi məsələnin optimallıq prinsipinə cavab verməsini tələb edir: onun hər hansı nüsxəsinin optimal həlli onun alt nüsxələrinin optimal həllərindən ibarət olmalıdır.

Paskal üçbucağını qurmaqla binom əmsallarının hesablanmasını optimallaşdırma məsələsi olmayan məsələyə dinamik proqramlaşdırma metodunun tətbiqi kimi qiymətləndirmək olar.

Fibonaççi ardıcıllığı aşağıdakı rekurrent münasibət ilə müəyyən edilən məşhur ədədlər ardıcıllığıdır: $F(n) = F(n-1) + F(n-2)$, burada $F(0) = 0$ və $F(1) = 1$, $n \geq 2$. Fibonaççi ədədlərinin hesablanması üçün sadə rekursiv alqoritmi (rekurrent münasibətin birbaşa istifadəsilə) tətbiq etmək olardı, lakin bu, eksponensial vaxt effektivliyinə gətirib çıxarardı. Dinamik proqramlaşdırma, artıq həll edilmiş alt məsələlərin nəticələrini saxlayan ***memoizasiyadan*** istifadə edərək xətti vaxtda bu problemi həll etməyə imkan verir.

Qrafın bütün təpələr cütləri arasında ən qısa yolların tapılması məsələsinin həlli üçün Floydun alqoritmi dinamik proqramlaşdırma metodunun tətbiqinə əsaslanır. Dinamik proqramlaşdırma metodundan istifadə edərək çanta məsələsinin həlli, bu metodun mürəkkəb kombinator optimallaşdırma məsələlərin həlli üçün tətbiq edilməsinə misaldır.

"***Xəsis***" (greedy – xəsis, acgöz) metod, hər biri ilkin məsələnin tam həllinə nail olunana qədər qismən qurulmuş həlli genişləndirən addımlar ardıcıllığı vasitəsilə problemin həllini qurmaqdan ibarətdir. Hər addımda məqbul, yerli olaraq optimal və yekun olan seçim edilməlidir. Pul dəyişdirmə məsələsinə (cange-making problem) baxaq: müəyyən bir ölkədə istifadə edilən sikkələrin mövcud nominallarının $d_1 > d_2 > \ldots > d_m$ ən kiçik sayından istifadə etməklə n məbləğini necə ödəmək olar? Məsələn, ABŞ-da d_1=25 sent (quarter), d_2=10 sent (dime), d_3=5 sent (nickel) və d_4=1 sent (penny) geniş istifadə olunur. 25, 10, 5 və 1 sentlik nominalında mövcud sikkələrdən müəyyən bir məbləği, məsələn, 63 qəpik yığmaq məsələsinin həllinə "xəsis" metodundan istifadə edək. Məsələnin həllinin hər bir mərhələsində "xəsis" alqoritm bu mərhələdə lokal olaraq optimal olan variantı seçir. "Xəsis" alqoritm hər dəfə qalan məbləğin dəyərindən çox olmayan ən yüksək nominallı sikkənin seçilməsi, onu sikkələr siyahısına əlavə etmək və qalan məbləğdən qiymətini çıxmaqdan ibarətdir. Alqoritm tez bir zamanda məsələnin həllinə gətirib çıxaracaq: 25 sentlik 2 dənə, 10 sentlik 1 dənə, 1 sentlik 3 dənə. "Xəsis" alqoritmlər yaxşı həllər əldə etməyə imkan verir, lakin bu həllər həmişə fərqli şərtlərdə optimal olmur. Məsələn, 11, 5 və 1 sentlik sikkələrdən 15 sentlik bir məbləğ toplamaq lazımdırsa, onda "xəsis" alqoritm belə nəticə verəcək: 11 qəpiklik 1 dənə. və 4 dənə 1 qəpiklik (cəmi 5 qəpik). Ancaq eyni zamanda 3 dənə 5 qəpiklik də kifayət edərdi.

"Xəsis" alqoritmlərə misal kimi çəkili əlaqəli qrafın minimal bünövrə ağacının (minimum spanning tree) tapılması üçün Prim və Kruskal alqoritmləri və çəkili qrafın bir təpəsindən digərlərinə ən yaxın məsafənin tapılması məsələsi (single-source -paths problem) üçün Deykstra alqoritmini göstərmək olar.

Qayıdışla axtarış (ingiliscə backtracking) müəyyən bir M çoxluğunda bütün mümkün variantların tam axtarışını tələb edən problemin həlli yollarının tapılmasının ümumi üsuludur. Bir qayda olaraq, bu kimi suallar verən məsələləri həll etməyə imkan verir: " Bütün mümkün variantları sadalayın..." , "Neçə yol var...", "Bir yol varmı...", "Obyekt varmı..." və s. *Backtracking* termini 1950-ci ildə amerikalı riyaziyyatçı Derrik Henri Lemer tərəfindən istifadə edilmişdir.

Məsələnin qayıdışla axtarış metodundan istifadə edərək həlli qismən həllin ardıcıl genişlənməsinə gətirilir. Əgər belə bir genişlənmə növbəti mərhələdə həyata keçirilə bilməzsə, onda daha qısa bir qismən həll yoluna qayıdılır və axtarış davam etdirilir. Bu alqoritm məsələnin bütün həll yollarını (əgər onlar varsa) tapmağa imkan verir. Metodunu

DOI:10.30546/gdu.1.2024

sürətləndirmək üçün hesablamaları elə təşkil etməyə çalışırlar ki, uyğun olmayan variantları mümkün qədər tez müəyyən etsinlər. Çox vaxt bu, məsələnin həllini tapmaq üçün vaxtı əhəmiyyətli dərəcədə azalda bilər.

Qayıdışla axtarış alqoritmindən istifadənin klassik nümunəsi səkkiz şaxmat vəzirləri haqqında məsələsidir. Onun qoyuluşu aşağıdakı kimidir: "Standart 64 kvadratlıq şahmat taxtasına 8 vəziri elə düzün ki, onların heç biri digərinin hücumuna məruz qalmasın". Əvvəlcə lövhəyə bir vəzir qoyurlar, sonra isə hər bir sonrakı vəziri yerləşdirməyə çalışırlar ki, artıq quraşdırılmış vəzirlər onu vura bilməsin. Əgər növbəti mərhələdə belə bir quraşdırma etmək mümkün deyilsə, onda bir addım geri qayıdırlar və əvvəl yerləşdırılmış vəziri başqa yerə qoyurlar.

Bundan əlavə, qayıdışla axtarış metodu bir çox digər axtarış məsələlərini həll etməyə imkan verir. Məsələn, ondan istifadə edərək, verilmiş M çoxluğunun bütün alt çoxluqlarını, yerləşdirmələrini (permutasiya), təkrarsız yerdəyişmələrini (k-permutasiya), birləşmələrini (combinizon) əldə etmək olar. Bununla belə, hətta kiçik ölçülü məsələ həllinin tapılması çox vaxt tələb edə bilər.

Alqoritmin hazırlanması proqramlaşdırma prosesinin mühüm mərhələsidir və proqramçılara mürəkkəb məsələləri başa düşülən, praktiki olaraq əldə edilə bilən addımlar ardıcıllığına çevirməyə imkan verir.

Əksər hallarda verilmiş məsələ bir neçə üsulla həll edilə bilər. Məsələnin həlli üçün xüsusi metodun seçimi adətən aşağıdakı meyarlara uyğun olaraq aparılır:

- məsələnin həlli üçün optimal vaxtın təmin edilməsi;
- mövcud resurslardan (yaddaşdan) optimal istifadənin təmin edilməsi;
- hesablamaların tələb olunan dəqiqliyinin təmin edilməsi;
- minimum xərclər;
- standart alt proqramlardan istifadə etmək imkanı.

Ədəbiyyat

1. Макконелл Дж. Основы современных алгоритмов. 2-е дополненное издание.М.: Техносфера, 2004.
2. Кнут Д. Искусство программирования. Тома 1, 2, 3. 3-е изд. Пер. с англ. : Уч.пос. — М.: Изд. дом "Вильямс", 2003.
3. Левитин А.В. Алгоритмы. Введение в разработку и анализ. М.: Издателький дом "Вильямс" 2008 год.
4. Кормен Т., Лейзерсон Ч., Ривест Р.Алгоритмы:построение и анализ.М.МЦНМО,2013
5. Ахо А., Хопкрофт В., Ульман Д.Структуры данных и алгоритмы. М.:«Вильяме»,2000
6. Феофанов О.Б. Алгоритмы и структуры данных (учеб. пособие), Изд-тво Томского политехнического университета, 2014.
7. Бхаргава А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. - СПб.: Питер, 2017. - 288 с.: ил. - (Серия «Библиотека программиста»).
8. Левитин А., Левитина М. Алгоритмические головоломки [Электронный ресурс]— М. :Лаборатория знаний, 2019.
9. https://ru.wikipedia.org/
10. https://mgutunn.ru/work/1307036/Metody-razrabotki-algoritmov
11. https://proglib.io/p/25-algoritmov-dinamicheskogo-programmirovaniya-kotorye-dolzhen-znat-kazhdyy-programmist-2023-05-30

DOI:10.30546/gdu.1.2024

МЕТОДЫ РАЗРАБОТКИ АЛГОРИТМОВ

Бабаева Шакяр Маарифовна

Резюме

Алгоритмы имеют первостепенное значение как в научной так и в технической сфере. Основной целью университетского образования при преподавании курса, посвященного разработке и анализу алгоритмов считается выработка у студентов навыков самостоятельного решения поставленных задач.

Метод проектирования алгоритма – это универсальный подход, применяемый для алгоритмического решения широкого круга задач, относящихся к различным областям вычислительной техники. Методы проектирования алгоритмов можно считать универсальным средством решения задач. Их применение не ограничивается только традиционными задачами вачислительной техники и математики. В данной статье исследуется важность изучения алгоритмов и методов их разработки, дается обзор существующих основных методов разработки алгоритмов и также приводятся некоторые примеры их применения для решения классических задач вычислительной техники.

ALGORITHM DEVELOPMENT METHODS

Babayeva Shakar Maarif

Summary

Algorithms are of paramount importance in both the scientific and technical fields. The main goal of university education when teaching a course devoted to the development and analysis of algorithms is to develop students' skills in independently solving assigned problems.

The algorithm design method is a universal approach used for algorithmically solving a wide range of problems related to various fields of computer technology. Algorithm design methods can be considered a universal means of solving problems. Their application is not limited only to traditional problems of computer technology and mathematics. This article explores the importance of learning algorithms and methods for developing them, provides an overview of the existing basic methods for developing algorithms and also provides some examples of their application to solve classical problems in computer science.

Rəyçi: Aslanov Əkrəm Əhməd oğlu
İnformatika və Cəbr kafedrasının 15 fevral 2024-cü il tarixli iclasın 09 saylı protokolu
Daxil loma tarixi 15 fevral 2024-cü il