Марат Оспанов атындағы Батыс Қазақстан мемлекеттік медицина университеті

Студенттің өзіндік жұмысы

Тақырыбы: Алгоритмдеу
Курс: I
Тобы: 111 А
Орындаған: Әбиболла И.С
Тексерген: Сағынбаева С.С
Күні: 12.02.2016ж

Ақтөбе-2016ж

Жоспар:

- I.Кіріспе
- II.Негізгі бөлім
- 1.Алгоритмдеудін тиімділігін талдау және бағалау
- 2.Алгоритм қасиеттері

Кіріспе

- **Алгоритм ұғымы**
- **Алгоритм** алгорифм (ағылшынша: algorithm, algorismus — Әл-Хорезмидің атынан шыққан) — бастапқы берілген мәліметтермен бір мәнде анықталатын нәтиже алу үшін қай амалды (жұмысты) қандай ретпен орындау қажеттігін белгілейтін есептерді (мәселелерді) шешу (математикалық есеп-қисаптар орындау, техникалық объектілерді жобалау, ғылыми-зерттеу жұмысын жүргізу т.б.) тәсілдерінің дәл сипаттамасы. Алгоритм — математика мен кибернетиканың негізгі ұғымдарының бірі. Агоритмді орындау алгоритмдік процесс деп аталады.
- **алгоритм** дегеніміз – жеке қадамдардан тұратын, формальды түрде жазылған реттелген нұсқаулар тізбегі. Алгоритм сөзі IX ғасырда өмір сүрген ұлы өзбек математигі **Әл-Хорезмидің** атымен аталған жазудың латындық формасы. **Әл-Хорезми** бірінші рет арифметикалық амалдарды орындаудың ережелерін тұжырымдаған ғалым. Алгоритм ұғымы кез-келген программа құру кезінде негізгі орын алады, себебі программа – енгізілген берілгендерді өңдеу үшін арнайы және қатаң түрде қандай да бір программалау тілінде дайындалған алгоритм. Кез-келген алгоритм қандай да бір орындаушыға негізделген. Орындалған командалар жиынтығы орындаушының командалар жүйесі болып табылады. Орындаушы ретінде – адамдар және техникалық құрылғылар, яғни роботтар, компьютерлер және автоматтар болуы мүмкін.

1.Алгоритмдеудін тиімділігін талдау және бағалау

- Алгоритм қандай да бір машинада командалардың жинағы түрінде орындалады. Бір есепті орындауға арналған екі немесе бірнеше алгоритмдердің орындалу жылдамдығын салыстыру үшін қолданылатын критерий (өлшем) **жүйелік тиімділік** деп аталады. Бір компьютерде бірдей деректер жинағымен бұл алгоритмдерді орындату арқылы олардың орындалуына кеткен салыстырмалы уақытты анықтауға болады.Ол үшін ішкі жүйелік сағат қолданылады.
- **Ішкі жүйелік сағатты қолданып уақытты бағалау** бұл бір ғана есептің орындалуына арналған алгоритмдердің әрқайсысының жүйелік тиімділігінің өлшемі болады.
- Кейбір алгоритмдердің орындалуында жадқа қойылған шектеулер проблема тудырады. Орындалу барысында ұзақ уақыт сақтау үшін бастапқы көлемді қысу қажет болады.**Қандай да бір алгоритм пайдаланатын ішкі жадтың салыстырмалы санының өлшемі** — бұл **кеңістіктің тиімділігі** (space efficiency). Бұл критерий алгоритмді қандай типті компьютер орындай алатынын және алгоритмнің толық жүйелік тиімділігін көрсетеді. Жаңа компьютерлік жүйелердің жадтарының көлемінің ұлғаюна байланысты бұл критерий маңызды болмай қалды.
- Үшінші тиімділік критерийі — бұл **есептеу тиімділігі** (computational efficiency), ол алгоритмнің ішкі құрылымын қарастырады, оның жасалуын және алгоритмде қолданылатын итерациялар мен меншіктеу операторларын салыстыратын тесттердің санын да талдайды.

Іздеу алгоритмдерінің күрделілігін бағалау -Big-0 нотациясы

Әдетте күрделі алгоритмнің ен жаман және ең жақсы жағдайлары үшін әртүрлі есептеу тиімділігі болады, сондықтан әр жағдай үшін есептеу тиімділігін өлшеу үшін, яғни алгоритмнің орындалу уақытының өлшемін анықтау үшін Big-0 нотациясын есептейміз.

Big-0 нотациясы. Алгоритмнің есептеу тиімділігі өңделетін деректердің санымен, яғни алгоритмдегі басты операциялардың санымен анықталады. Бұл операциялар деректің типіне, санына, реттелуіне тәуелді болады. Көп жағдайда сұрыптау алгоритмдерінде (тізімдер мен бұтақтарда) алгоритм тиімділігін өлшемі ретінде салыстыру қолданылады. Ол массив элементтерінің санына тәуелді.

Іздеу алгоритмдерінің күрделілігі. Big-O-бағалау алгоритмнің орындалу уақытының өлшемін (runtime) береді. Әдетте алгоритмнің жақсы және нашар жағдайлар үшін есептеу тиімділіктері әртүрлі , сондықтан әр нақты жағдай үшін Big-O мәні есептеледі. Егер алгоритм реті —*0(1)* болса, оның реті коллекциядағы элементтер санына тәуелсіз болады. Бұл алгоритм тұрақты уақыт бірлігі ішінде орындалады, мысалы егер массив соңын көрсететін индекс сақталса, массив элементіне мән меншіктеу реті *0(1)* болады.

Алгоритм *O(п)* сызықты (linear). Оның күрделілігі тізім размеріне пропорционал.

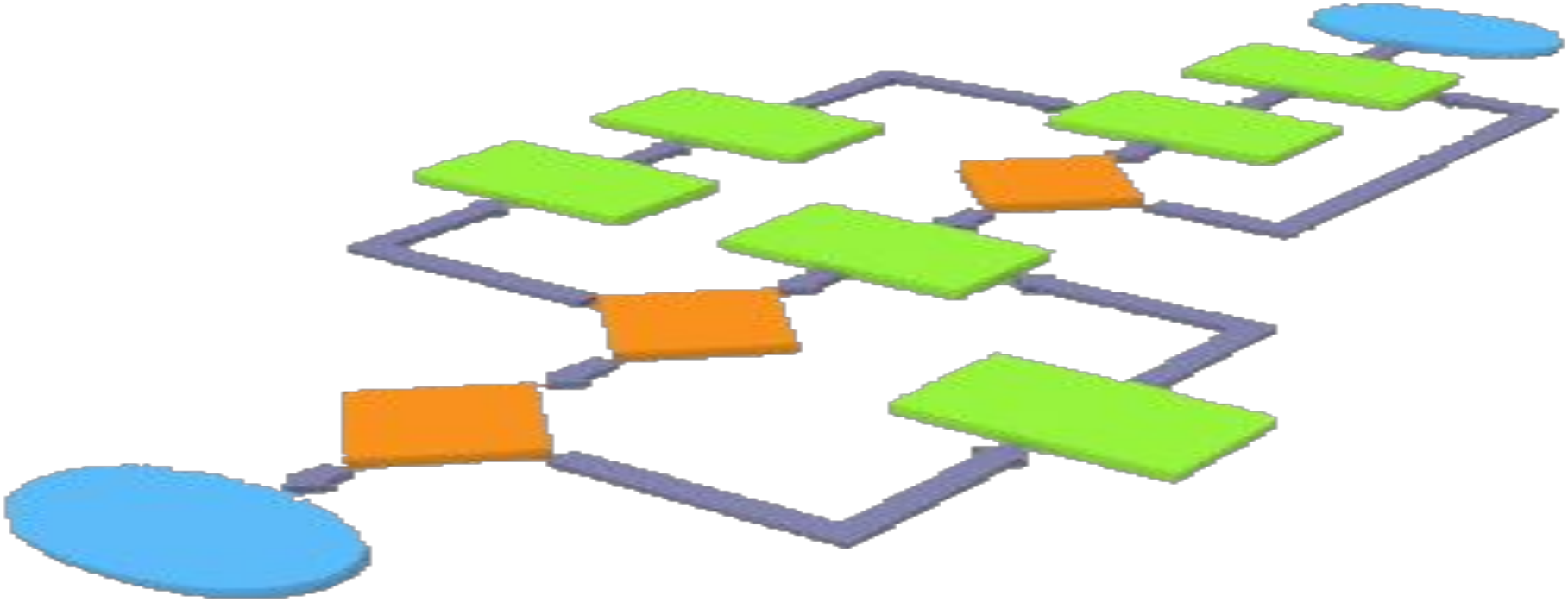
Реті $log_2 n$ болатын алгоритмдер логарифмдік (logarithmic) деп аталады. Мұндай күрделілік тізімдерді бірнеше рет ішкі тізімдерге 1/2, 1/4, 1/8 етіп бөлгенде туындайды. Мысалы бинарлық бұтақтарда іздеу алгоритмдерінің күрделілігі орташа және нашар жағдайлар үшін $O(log_2 n)$ болады.

Реті $O(п^2)$ болатын алгоритмдерквадраттық (quadratic) деп аталады. Шағын n үшін ғана практикада қолданылады. *п* екіге артқан сайын алгоритмнің орындалу уақыты 4-ке артады.

Реті $O(n^3)$ болатын алгоритмдер өте баяу орындалады, кубтық (cubic) уақытты қажет етеді. *п* екіге артқан сайын алгоритмнің орындалу уақыты 8 есе артады. Оның мысалына графтарға қолданылатын реті $O(п^3)$ болатынУоршел алгоритмі жатады.

Күрделілігі $O(2^п)$ тең алгоритмде экспоненциальды күрделілік (exponential complexity) болады. Өте баяу орындалатындықтан өте аз *п* үшін қолданылады.

Математика
Русский язык
География
Литература
АЛГОРИТМ
Физика
Английский язык
История
Технология



Алгоритмнің графиктік түрде кескінделуі

Алгоритмнің графиктік түрде кескінделуі – кең таралған әдіс. Бұл – жазудың түсінікті, анық, көрнекі түрі болып табылады. Алгоритмдерді графиктік жолмен жазудың мемлекеттік стандарты анықталған.