ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ

«Л.Н. ГУМИЛЕВ АТЫНДАҒЫ ЕУРАЗИЯ ҰЛТТЫҚ УНИВЕРСИТЕТІ» КЕАҚ

Студенттер мен жас ғалымдардың
«ǴYLYM JÁNE BILIM - 2024»
XIX Халықаралық ғылыми конференциясының
БАЯНДАМАЛАР ЖИНАҒЫ

СБОРНИК МАТЕРИАЛОВ
XIX Международной научной конференции
студентов и молодых ученых
«ǴYLYM JÁNE BILIM - 2024»

PROCEEDINGS
of the XIX International Scientific Conference
for students and young scholars
«ǴYLYM JÁNE BILIM - 2024»

2024
Астана

УДК 001
ББК 72
G99

«ǴYLYM JÁNE BILIM – 2024» студенттер мен жас ғалымдардың XIX Халықаралық ғылыми конференциясы = XIX Международная научная конференция студентов и молодых ученых «ǴYLYM JÁNE BILIM – 2024» = The XIX International Scientific Conference for students and young scholars «ǴYLYM JÁNE BILIM – 2024». – Астана: – 7478 б. - қазақша, орысша, ағылшынша.

ISBN 978-601-7697-07-5

Жинаққа студенттердің, магистранттардың, докоранттардың және жас ғалымдардың жаратылыстану-техникалық және гуманитарлық ғылымдардың өзекті мәселелері бойынша баяндамалары енгізілген.

The proceedings are the papers of students, undergraduates, doctoral students and young researchers on topical issues of natural and technical sciences and humanities.

В сборник вошли доклады студентов, магистрантов, докторантов имолодых ученых по актуальным вопросам естественно-технических и гуманитарных наук.

УДК 001
ББК 72
G99

ISBN 978-601-7697-07-5

Алынған, осы u және W функцияларға қатысты нәтижені және мұнда енді біз осы түрлендірудегі екінші көбейткіште интегралдау шектерін кеңейтіп, $z = x_k$ ауыстыруын жасап, sup алып және бірінші көбейткішке$p \le q$ жағдайына сәйкес Йенсен теңсіздігін қолдана отырып, $J_1 \ll A^q \|f\|_{p,w}^q$теңсіздігі орындалады. Енді біз J_2 өрнегін жоғарыдағы J_1-ді бағалағандағыдай, Теорема – А бойынша $\|Hf\|_{q,v}^q \ll A^q \|f\|_{p,w}^q$болады.Демек, $J_2 \ll A^q \|f\|_{p,w}^q$
болады.

$$\|Tf\|_{q,v} \ll A\|f\|_{p,w}$$

Сәйкесінше, біз есептеген интегралымыздан және табылған J_1 және J_2 теңсіздіктерден біз келесі қорытындыға келеміз:

$$\|Tf\|_{q,v} \ll A\|f\|_{p,w}.$$

Яғни T операторы $L_{p,w}$ кеңістігінен $L_{q,v}$ кеңістігіне $p \le q$ жағдайында шенелген болады. Сонымен қатар, $\|T\| \ll A$және $\|T\| \gg A$ болады, онда $\|T\| \approx A$ және мұндағы $A = \sup_{z \in I} A(z) < \infty$ болады. Жеткіліктілігінің орындалатынын көрсеттік. Теорема толық дәлелденді.

Пайдаланылған әдебиеттер тізімі

1. AbylayevaA.M.,OinarovR. and Seilbekov B. Boundedness and compactness of a class of integral operators with power and logarithmic singularity when $p \le q$. // Journal of Inequalities and Applications – 2022, – №.23.

2. AbylayevaA.M. Boundedness and compactness for a class of fractional integration operators of Weyl type. // Eurasian Mathematical. J. – Volume 7, – 2016, – No.1, – C. 9–27.

3. Andersen K.F. andSawyer E.T. WeightednorminequalitiesfortheRiemann–LiouvilleandWeylfractionalintegraloperators. // Trans. Amer. Math. Soc, – Volume 308, 1988, – C. 547–558.

4. MeskhiA. Solution of some weight problems for the Riemann–Liouville and Weyl operators. // Georgian Math.J. – Volume 5, 1998, – No.6, – C. 565–574. – Volume 106, 1989, – C. 727–733.

5. Prokhorov D.V. On the boundedness and compactness of a class of integral operators. // J. London Math. Soc, – Volume 61, 2000, – No.2, – C. 617–628.

6. Прохоров Д.В. и Степанов В.Д. ВесовыеоценкидляоператоровРимана–Лиувилля и ихприложений. // ТрудыМатематическогоинститута РАН, – Том 243, 2003, – С. 289–312.

УДК 519.1

КОМБИНАЦИЯЛЫҚ ӘДІСТЕР: РЕКУРСИЯ ЖӘНЕ ДИНАМИКАЛЫҚ БАҒДАРЛАМАЛАР

Серік Қаратай Бақтыбайұлы

karatai_serik@mail.com

Л.Н.Гумилев атындағы ЕҰУ 7М05401 – Математика және статистика мамандығының 2-курс магистранты, Астана, Қазақстан

Ғылыми жетекшісі – Канкенова Аягоз Мелисовна

Бүкіл әлемде, атап айтқанда, Қазақстанда бағдарламалау бойынша олимпиадалық қозғалыстың өзектіленуі мен күшеюіне байланысты студенттерді олимпиадаларға қатысуға

дайындау мәселесі барған сайын бәсекені жоғарылатуда. Мұндай іс-шаралар сенімді маман даярлаудың негізгі негізі болып табылады. Олимпиаданы бағдарламалау әдістері мен әдістерін білу бағдарламалық қамтамасыз етуді құру үшін берік негіз береді. Соңғы жылдары Қазақстандағы олимпиадалық қозғалыс айтарлықтай пропорцияға ие болды. Соның нәтижесінде оқушылардың білім деңгейі, біліктілігі айтарлықтай артты.

Қазақстанда көп олимиадаға дайындайтын үйірмелер жұмысы барысында автоматтандырылған on-line тексеру жүйелерін қолдану арқылы әр түрлі күрделілік деңгейіндегі есептерді шешуге оқыту жүргізіледі. Бұл қатысушылардың дайындығы мен жеке өсуін, жағдайды шарлау және алгоритмдерді тәжірибеде қолдану қабілетін бақылауға мүмкіндік береді. Сонымен қатар, студенттер әлемдік деңгейдегі олимпиадаларда қолданылатын on-line тестілеу жүйелерімен танысу мүмкіндігіне ие.

«Математикалық олимпиада есептері» атты оқу құралы оқу бағдарламасында кездеспейтін логикалық, шығармашылық, қызықты және қиындатылған есептерден құрастырылған. Оқушыны өз бетімен жұмыс істеуге икемдейді, сонымен қатар оның ғылыми тұрғыда ойлануына, танымдық және шығармашылық қабілеттерінің оянуына ықпал жасайды. Логикалық есептер оқушыны зерделілікке, тапқырлыққа, ұстамдылыққа, шапшаң есептеу қабілетін дамытуға, ойын ұштай түсіп, еңбектене білуге тәрбиелейді. Соның нәтижесінде оқушы математикалық біліктілігі мен дағдысын олимпиадада көрсете алады.

Математикадан олимпиада есептерін құрастыру әдетте математиканың әртүрлі салаларында қатысушылардың дағдыларын тексеретін қызықты және күрделі математикалық есептерді құру мүмкіндігін талап етеді. Мұнда олимпиадаға есептер құрастыруға көмектесетін бірнеше қадамдар берілген.

1. Тақырыпты таңдау. Сіз сынағыңыз келетін математика саласын анықтауыңыз керек (мысалы, алгебра, геометрия, комбинаторика және т.б.)

2. Формуласы. Біз мәселені түсінікті, бірақ сонымен бірге қызығушылықты тудыратын және әртүрлі тәсілдермен шешуге болатындай етіп тұжырымдаймыз.

3. Қиындық деңгейі. Олимпиадаға қатысушылардың деңгейін ескеріп, тапсырманың күрделілігінің сәйкес деңгейін таңдау керек.

4. Әртүрлілік. Қатысушылардың әртүрлі дағдылары мен ойлау тәсілдерін тексеру үшін әртүрлі сынақтар жасауға тырысыңыз.

5. Шешімді тексеру. Мәселенің бірегей шешімі бар екеніне және шешімді тексеруге болатынына көз жеткізіңіз.

6. Ерітінді дайындау. Барлық қажетті түсініктемелер мен шешу қадамдарын қамтитын мәселенің толық және түсінікті шешімін дайындаңыз.

7. Тестілеу. Оны олимпиадада қолданбас бұрын, оның жарамдылығын тексеру үшін мәселені әртүрлі қатысушыларға немесе сарапшыларға сынап көріңіз.

8. Бейімделу. Қажет болса, алдыңғы олимпиадалардағы кері байланыс пен тәжірибе негізінде тапсырманы бейімдеңіз.

Үйірмелер аясында студенттерді дайындаудың басым бағыттарының бірі комбинаторлық әдістерге оқыту болды. Комбинаторлық әдістерді қолдану әртүрлі алгоритмдер мен деректер құрылымдарын жақсы білуді, сонымен қатар жақсы бағдарламалау дағдыларын қажет етеді. Комбинаторлық есептердегі ең өзекті нәрсе әртүрлі санау және есептеу нұсқаларының ішінен оңтайлы шешімді іздеу және таңдау болып табылады, бұл бір уақытта бағдарламаларды құрастыру қабілетін жақсартатын оқыту.

Комбинаторлық әдістер негізінен эвристикалық сипатқа ие, есептердің әртүрлі кластары үшін де, жеке есептер үшін де жеке. Оның үстіне, мұндай әдіс неғұрлым нақтырақ айқын болса, мәселе компьютерде тиімдірек шешіледі.

Комбинаторлық бағдарламалау әдістері деп бағдарламаларды құрудың әдістерінің, құралдарының және технологияларының жиынтығын түсінеміз. Оларға мыналар жатады: дөрекі күш, рекурсия, динамикалық бағдарламалау және кері трек.

Мен динамикалық бағдарламалау әдістерін және қайталанатын формулаларды егжей-тегжейлі қарастырғым келеді, өйткені қателер мен келіспеушіліктердің ең көп саны осы әдістерді қолданумен байланысты.

Қайталанатын тәсіл және динамикалық бағдарламалау бір-бірін толығымен алмастыратын шешім әдістері болып табылады. Қайталанатын формуламен көрсетілген кез келген мәселені ұсынылған әдістердің кез келгенімен шешуге болады. Сондықтан әрбір қайталанатын мәселені шешу барысында кез келген әдісті қолданудың орындылығы туралы сұрақ туындайды. Өйткені, бағдарламаның орындалу жылдамдығы, оны оқудың қарапайымдылығы және компьютерлік ресурстарды пайдаланудың тиімділігі сәтті таңдауға байланысты.

Өздерін шақыратын функциялар рекурсивті деп аталады. Рекурсия рекурсивті түрде анықталған функциялардың есептеулерін өте қарапайым (циклдерді қолданбай) бағдарламалауға мүмкіндік береді. Мысалы, f(n) =n!факторлық функциясы белгілі: f(0) =1; f(n) =n*f(n – 1).

Бұл операция мәселені анық бірнеше кіші қосалқы тапсырмаларға бөледі, олардың әрқайсысының шешімі бастапқы есептің шешіміне әкеледі. Осылайша, іске қосылған функция тривиальды мәселені шешу үшін «кеңейеді» (факторлық үшін - n=1 немесе n=0 болған жағдай), содан кейін бастапқыға «жиырады». Бірақ мұндай әрекеттер динамикалық бағдарламалау әдісінің анықтамасымен толық сәйкес келетінін көреміз. Жалғыз айырмашылық «орналастыруда».

Осылайша, рекурсивті әдіс есептің қажетті түйіндері арқылы қанша рет болса, сонша рет өтеді. Динамикалық бағдарламалау әдісі, керісінше, барлық ішкі міндеттерді шешеді, тіпті бастапқы мәселені шешу үшін қажет емес, бірақ әрбір түйінді тек бір рет өтуге кепілдік беріледі.

Мысалы, Фибоначчи сандарының белгілі тізбегін қарастыратын болсақ, n-ші санды генерациялау үшін динамикалық бағдарламалау әдісі дәл n әрекетті орындайтынын көреміз - әрбір келесі санды дәл 1 рет тікелей іздеу. Бұл ретте рекурсияны қолдану бір сандарды жіберіп алмай, қайталап есептеуді талап етеді.

Сонымен қатар, ең үлкен ортақ бөлгішті іздеу кезінде қайталанатын алгоритм жақсы өнімділікке ие, бұл қайталанатын қоңыраулардың болмауына байланысты.

Зерттеу барысында «Ханой мұнарасы» классикалық тапсырма ойыны қарастырылды, ол келесідей: үш таяқша бар, олардың біріншісінде N сақина пирамидасы (төменгі жағында үлкен сақиналар, үстіңгі жағында кішірек сақиналар) бар.), сақиналарды басқа штангаға жылжыту керек. Сақиналарды өзекшеден өзекке ауыстыруға рұқсат етіледі, бірақ кішірек сақинаның үстіне үлкенірек сақинаны қоюға болмайды. Қажетті әрекеттер тізбегін генерациялайтын программа құрыңыз.

Бұл мәселені шешудің кем дегенде екі жолы бар екенін ескеріңіз: рекурсивті және рекурсивті емес (қатал күш). Бұл жағдайда рекурсивті әдіс айқынырақ, бірақ шешу барысында динамикалық бағдарламалау әдісін қолданғанда қайталанатын формуланы жүзеге асыру оңтайлы екені белгілі болады.

Мақалада [1] берілген тікбұрышты матрицаны пайдаланып латын шаршысын толтыру мәселесі шешілді. Таңдалған шешім әдісі тиімді, бірақ оңтайлы емес, кері қадаммен өрескел күш екенін ескеріңіз. Кейіннен біз бұл мәселені қайталанатын формулалар әдісі арқылы шештік, бұл бағдарламаның жұмыс уақытын айтарлықтай қысқартуға мүмкіндік берді.

Өкінішке орай, рекурсия және динамикалық бағдарламалау әрқашан оңтайлы жұмыс істей бермейді. Есептелген мәндерді есте сақтау арқылы айтарлықтай жылдамдатуға болмайтын ұзақ жұмыс істейтін рекурсивті алгоритмі бар есептің мысалын қарастырайық.

Саяхатшы әр түрлі тауарларды сату үшін қалаларды аралайды. Ол әрқашан сапарын бір қалада бастап, бір қалада аяқтайды. Бизнесмен барлық қалаларда тұру үшін бірдей ақша жұмсайды. Сондықтан қаладан қалаға жол жүру сомасы ғана қомақты. Қалалардың тізімі және олардың кейбір жұптары арасында жол жүру ақысы бар.

Саяхатшы сатушының міндеті - барлық қалаларды аралау (кем дегенде бір рет), саяхатқа мүмкіндігінше аз ақша жұмсап, кері қайтару.

қала ішінде жол жүру ақысыз;

екі қала арасындағы тікелей жол жүру екі бағытта бірдей шығындар;
құны – 1-ден 10000-ға дейінгі бүтін сан;
100 қаладан артық емес.
Жұп қалалар арасындағы жол жүру құны келесі форматта жазылған:
<Шығындар саны><Бастапқы қала>
<Қала><Қала><Құны>
<Қала><Қала><Құны>
<Қала><Қала><Құны>
...
Нәтиже келесі форматта жазылады:
<Маршруттағы қалалар саны>
<Қала><Қала><Қала> …
Қала бос орынсыз атымен көрсетілген. Қала атауының ұзындығы 30 таңбадан аспайды.

Бір қарағанда, мұндай мәселені шешу үшін ең оңтайлы аралық жолдарды сақтау жеткілікті, содан кейін біз бірден динамикалық алгоритмді аламыз. Мысалы, үш қаланы қарастырайық және әрбір үш қала үшін оны қандай ретпен өткен дұрыс екенін анықтаймыз. Содан кейін төрт қаланы түгел қарастырамыз. Төрттік үшін шешімдерді үш еселік және т.б. үшін белгілі шешімдер негізінде табуға болады. Бірақ шын мәнінде қалалардың ықтимал жиынтықтары мен мүмкін болатындар саны өте тез өсіп келеді. Мысалы, 100 қаланың 50-ін 100,891,344,545,564,193,334,812,497,256 жолмен таңдауға болады. Осылайша, аралық шешімдерді есте сақтаудың ешқандай жолы жоқ - олар тіпті n=100 үшін де өте көп, бірақ іс жүзінде n=106 есептерді шешу керек. Ұшақтағы саяхатшы сатушы мәселесі үшін сіз ақылға қонымды уақыт ішінде қолайлы шешімдерді табуға мүмкіндік беретін бірқатар эвристикалық идеяларды пайдалана аласыз. Бірақ ұшақтағы нүктелер жағдайында да, саяхатшы сатушы мәселесі полиномды түрде шешілмейтін болып қалады, яғни кез келген k дәрежелі C+nk полиномымен шектелген уақытта ең оңтайлы саяхатшы жолын табатын алгоритм жоқ. кез келген тұрақты.

Осылайша, есептерді шешудің рекурсивті әдісі алгоритмдік есептерді шешудің негізгі әдісі дерлік болып табылады. Динамикалық бағдарламалау идеяларымен, ашкөз алгоритмдермен және кесу идеясымен толықтырылған рекурсия бағдарламашылардың ауыр артиллериясына айналады. Бірақ рекурсивті функцияларды белгілеудің қысқалығы әрқашан оларды есептеудің жоғары жылдамдығын білдірмейтінін ұмытпауымыз керек. Және рекурсия жай ғана зиянды болатын бірқатар мәселелер бар (мысалы, графиктегі ең қысқа жолды есептеу мәселесі).

Қорыта келгенде бағдарламалау әдістерінің оңтайлылығын талдау барлық комбинаторлық есептерді шешудің бір оңтайлы алгоритмін табу мүмкін еместігін көрсетті, әрбір есеп жеке көзқарасты қажет етеді. Қазіргі уақытта шешімдерді оңтайландыру іздеуді логикалық қысқарту болып табылады. Зерттеу есептерді шешудің рекурсивті әдісі мен динамикалық бағдарламалау әдісінің бір-бірін толығымен алмастыратыны, бірақ мұндай ауыстыру әрқашан ұтымды бола бермейтінін анық растады.

Қолданылған әдебиеттер тізімі

1. Беллман Р. Динамическоепрограммирование. / М.: Изд-воиностраннойлитературы, 1960, 400 с.
2. Броницька Н.А., Дармосюк В.М., Хомічук Р., Бережецька В. Програмуваннякомбінаторнихзадачнаприкладілатинськихквадратів // «Науковізаписки. Серія: Педагогічнінауки». 2012. Випуск 109.
3. Методы программирования // Алгоритмы и методы [Электронный ресурс] — Режим доступа. — URL: http://algolist.manual.ru/maths/combinat/sequential.php (дата обращения: 23.04.2013).