Алгоритмични и комбинаторни задачи по графи и динамично програмиране (домашна работа за студенти от СУ, ФМИ)

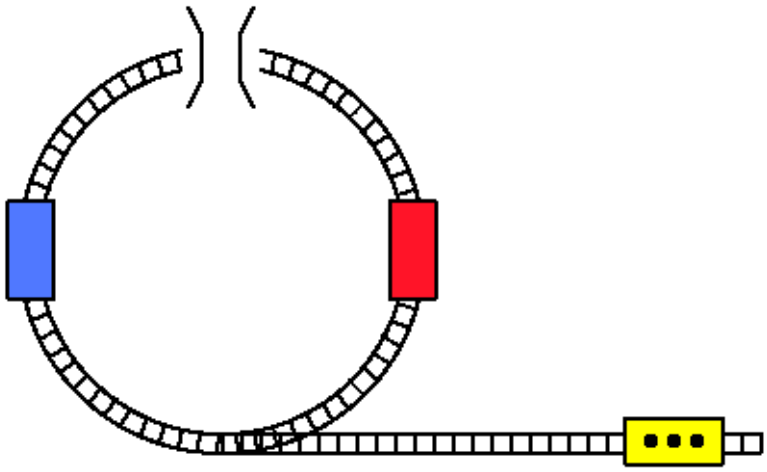
Име: ..... Факултетен № ..... Група: .....

Задача	1	2	3	Общо
получени точки				
максимум точки	*34*	*33*	*33*	*100*

Забележка 1: *Всички отговори трябва да бъдат обосновани подробно.*

Забележка 2: *Не предавайте идентични решения дори когато работите заедно: идентичните решения ще бъдат анулирани!*

Задача 1. Два вагона — червен и син — са разположени в двете половини на кръгова жп линия: червеният — в дясната половина; синият — в лявата. Достъп отвън до кръговата жп линия има откъм южната страна, по праволинеен участък, по който в този миг навлиза жълт локомотив.



Машинистът трябва да размени местата на вагоните: червеният да отиде в лявата половина на кръговата линия, а синият — в дясната, след което локомотивът трябва да отпътува сам. Локомотивът може да тегли и да бута вагоните — всеки поотделно или двата едновременно. Над северната част на жп линията има нисък мост. Локомотивът може да премине под моста, но вагоните не могат, защото са твърде високи.

Търси се решение с най-малък брой ходове. Ход наричаме всяко преминаване на локомотива от един участък в друг. Приемаме, че железопътната линия има три участъка:
— източен — където се намира отначало червеният вагон;
— западен — където се намира отначало синият вагон;
— южен — праволинейният участък, по който пристига и заминава локомотивът.

Подобни задачи се решават с компютърни програми, които извършват някакъв вид търсене в пространството на състоянията. В това домашно не се изисква писане на програмен код, а само подготвяне на задачата за програмиране, тоест съставяне на математически модел.

Пространство на състоянията се нарича множеството от всички възможни състояния. В тази задача състояние означава всяко разположение на локомотива и вагоните. Има значение в кой от трите участъка на жп линията се намират вагоните и локомотивът и как са подредени, когато два или три от тях са в един и същ участък.

а) Пространството на състоянията се представя чрез граф: върховете са състоянията, а ребрата са преходите между състоянията. Нарисувайте едно произволно избрано състояние и го свържете чрез ребра с всички състояния, до които може да се стигне от него за един ход. Изобразете състоянията чрез картинки с жп линията, вагоните и локомотива. **(2 точки)**

б) Какъв вид граф е по-подходящ за тази задача — ориентиран или неориентиран? Отговорът да се обоснове! **(2 точки)**

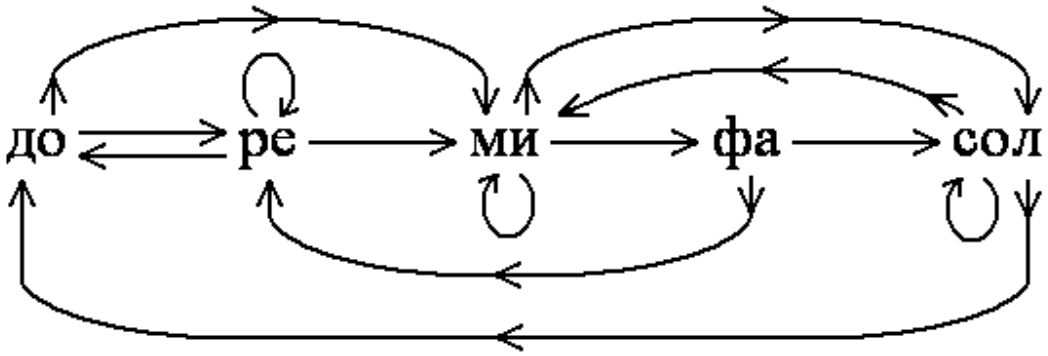
в) Рисуването на състоянията е нагледно, но не е удобно: отнема твърде много време. Измислете кратък, но ясен начин за записване на състоянията. Обяснете го в общия случай и го онагледете с чертежа от подусловие "а": до състоянията напишете кодовете им. **(4 точки)**

г) Пресметнете броя на състоянията по най-кратък начин. Колкото по-малко случаи се разглеждат в решението, толкова повече точки носи то. **(12 точки)**

д) Решете задачата за разместване на вагоните с най-малък брой ходове на локомотива. Опишете решението като редица от състояния. Всяко състояние да притежава пореден номер и да бъде записано кратко — със своя код. **(12 точки)**

е) Как се решава тази задача — чрез търсене в ширина или чрез търсене в дълбочина? Отговорът да се обоснове! **(2 точки)**

Задача 2. На картинката е показан граф, по който могат да се съчиняват различни мелодии: започваме от произволен връх на графа и последователно преминаваме през други върхове, като се движим по ребрата в указаната посока. Ако от текущия връх излизат няколко ребра, сами избираме по кое от тях да продължим обхождането.



По този начин може да се съставят различни мелодии с произволно избрана дължина, например следната мелодия от 24 ноти (първия куплет от песента "Тръгнал кос"):

сол–ми–ми–фа–ре–ре–до–ре–ми–фа–сол–сол–сол–сол–ми–ми–фа–ре–ре–до–ми–сол–сол–до.

Подобен подход може да бъде използван в компютърни програми за създаване на музика. Програмата избира началния връх на графа случайно и на всяка стъпка пак по случаен начин избира едно от ребрата, излизащи от текущия връх. Затова всяко изпълнение на програмата поражда различна мелодия. Броят на мелодиите расте бързо с увеличаване на дължината им.

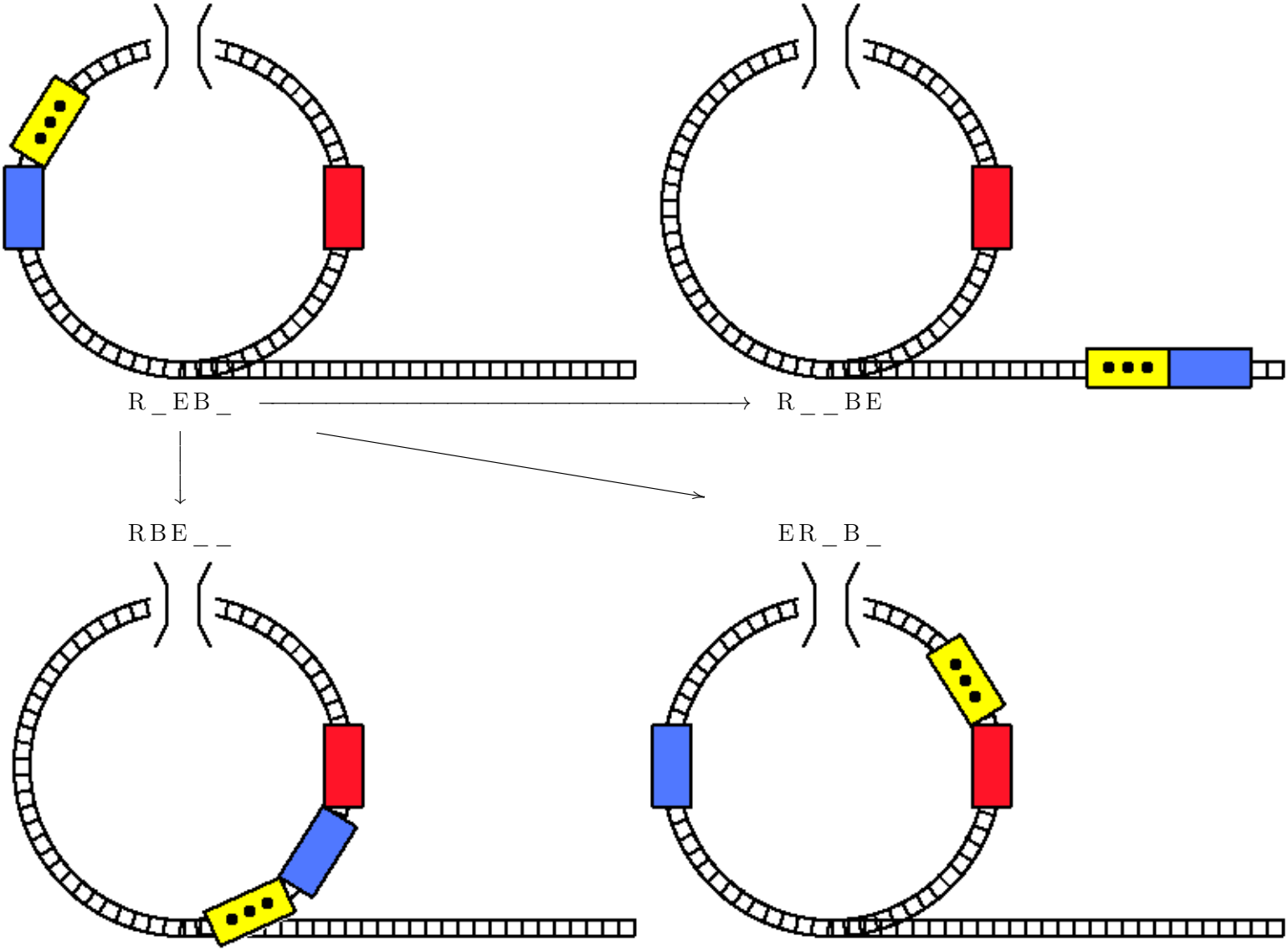
Колко мелодии от девет ноти можем да съставим по описания начин от графа на картинката? Задачата да се реши чрез степенуване на матрицата на съседство на графа. **(33 точки)**

Задача 3. Да се реши задача 2 с помощта на динамично програмиране. **(33 точки)**

РЕШЕНИЯ

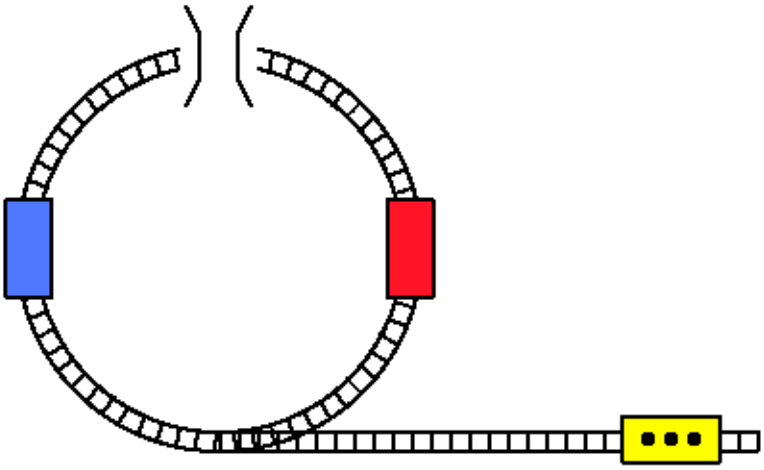
Задача 1. По принцип пространството на състоянията се представя чрез ориентиран граф, тъй като в общия случай преходите между състоянията са еднопосочни. В тази задача обаче преходите са двупосочни: за всяко възможно преместване на локомотива и вагоните обратното преместване също е възможно. Затова е по-добре да използваме неориентиран граф.

Графът е твърде голям, за да го начертаем целия, ето защо на чертежа по-долу показваме само един връх и излизащите от него ребра: от върха R _ E B _ излизат точно три ребра.

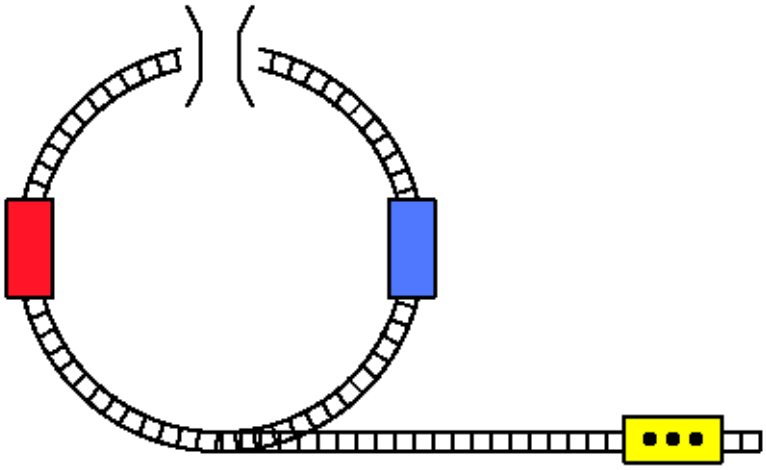


Всяко състояние е кодирано с низ от пет знака — латинските букви R, B, E и две долни черти. Латинските букви R, B и E означават съответно червения вагон, синия вагон и локомотива (от първите букви на английските думи *red*, *blue* и *engine*). Долните черти делят низа на три части: първата част описва източния участък от жп линията; втората — западния; третата — южния. Първите две части описват реда на вагоните и локомотива от север на юг (отгоре надолу), а третата част — от изток на запад (отдясно наляво).

Кодирането е биекция, затова състоянията са колкото низовете с пет знака от този вид. Низовете са пермутации с повторение, защото между знаците има еднакви — двете долни черти. Броят на пермутациите е $\widetilde{P}_5^{2;1;1;1} = \dfrac{5!}{2!1!1!1!} = 60$. Тоест съществуват общо 60 състояния и графът има 60 върха. (Дали те всички са достижими от началния връх, е отделен въпрос.)



R _ B _ E (начално състояние)



B _ R _ E (крайно състояние)

За да достигнем с най-малко ходове от даденото начално до желаното крайно състояние, трябва да намерим най-къс път от върха R _ B _ E до върха B _ R _ E (под дължина на пътя разбираме броя на ребрата, които пътят съдържа). Най-къс път се намира чрез търсене в ширина. По този начин намираме следното решение (то съдържа 20 хода — най-малкия възможен брой):

1) R _ B _ E;
2) R _ B E _ ;
3) R E _ B _ ;
4) _ B E R _ ;
5) _ B _ R E;
6) _ B E _ R;
7) E _ B _ R;
8) _ E B _ R;
9) _ _ R B E;
10) _ E B R _ ;
11) R B E _ _ ;
12) R _ E B _ ;
13) R _ _ B E;
14) R _ E _ B;
15) E R _ _ B;
16) _ R E _ B;
17) _ R _ B E;
18) _ R E B _ ;
19) B E _ R _ ;
20) B _ R E _ ;
21) B _ R _ E.

Тази идея — търсене на решение чрез обхождане на пространството на състоянията — е приложима и за други игри, поради което се използва в най-различни компютърни програми. Само че обхождането става много бавно, когато пространството на състоянията е голямо. За съжаление, повечето интересни игри (например шахматът) имат пространство на състоянията с астрономически размери. В такива случаи пълното изчерпване е неприложимо на практика. Вместо това се използват познанията за конкретната игра (да кажем, набор от заучени позиции или оценка на материалното преимущество в шахмата). Така задачата се свежда до обхождане на малка част от пространството на състоянията, а това може да се извърши за приемливо време. Нужно е обаче формализиране на знанията за играта, т.е. математическа теория. Създаването ѝ може да се окаже непосилно; тогава прибягваме към евристики. В дадена шахматна позиция може например да има проста тактика, гарантираща победа, макар и не с най-малко ходове. Тоест жертваме оптималността на решението, за да ускорим алгоритъма за търсене.

В задачата за двата вагона не възникват такива трудности: графът има само 60 върха, тоест пространството на състоянията е малко и търсенето в ширина е достатъчно бързо.

Търсенето в дълбочина е неподходящо тук: то не гарантира намирането на най-къс път.

Задача 2. Всяка мелодия представлява път в графа. Мелодиите с девет ноти съдържат девет върха, следователно осем ребра. Тоест тези мелодии съответстват на пътища с дължина 8, затова броят им е равен на сбора от елементите на A^8, където A е матрицата на съседство на дадения граф.

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{matrix} до \\ ре \\ ми \\ фа \\ сол \end{matrix}$$
$$до \ ре \ ми \ фа \ сол$$

Осмата степен на матрицата A пресмятаме най-бързо чрез трикратно повдигане на квадрат:

$$A^2 = \begin{pmatrix} 1 & 1 & 2 & 1 & 1 \\ 1 & 2 & 3 & 1 & 1 \\ 1 & 1 & 2 & 1 & 3 \\ 2 & 1 & 2 & 0 & 1 \\ 1 & 1 & 3 & 1 & 2 \end{pmatrix}, \quad A^4 = \begin{pmatrix} 7 & 7 & 14 & 5 & 11 \\ 9 & 10 & 19 & 7 & 15 \\ 9 & 9 & 20 & 7 & 15 \\ 6 & 7 & 14 & 6 & 11 \\ 9 & 9 & 19 & 7 & 16 \end{pmatrix}, \quad A^8 = \begin{pmatrix} 367 & 379 & 790 & 289 & 623 \\ 501 & 518 & 1079 & 395 & 851 \\ 501 & 517 & 1080 & 395 & 851 \\ 366 & 379 & 790 & 290 & 623 \\ 501 & 517 & 1079 & 395 & 852 \end{pmatrix}.$$

Сборът 14928 на елементите на последната матрица представлява отговорът на задачата: в дадения граф има 14928 пътя с дължина 8, т.е. съществуват 14928 мелодии от девет ноти. Отделните събираеми съответстват на мелодии с различно начало и край; например числото 379, което се намира в първия ред и втория стълб на матрицата A^8, означава, че има 379 мелодии, които започват с *до*, завършват с *ре* и съдържат още седем ноти.

Задача 3. Броя на мелодиите от девет ноти можем да намерим и с динамично програмиране. С $do(n)$ означаваме броя на мелодиите от n ноти, първата от които е *до*. По подобен начин определяме функциите $re(n)$, $mi(n)$, $fa(n)$ и $sol(n)$.

Начални условия: $do(1) = re(1) = mi(1) = fa(1) = sol(1) = 1$, защото от една дадена нота можем да съставим само една мелодия, без да повтаряме нотата.

Даденият граф се описва чрез следната система от рекурентни уравнения:

$$\left| \begin{aligned} do(n+1) &= re(n) + mi(n) \\ re(n+1) &= do(n) + re(n) + mi(n) \\ mi(n+1) &= mi(n) + fa(n) + sol(n) \\ fa(n+1) &= re(n) + sol(n) \\ sol(n+1) &= do(n) + mi(n) + sol(n). \end{aligned} \right.$$

Няма нужда да решаваме системата: числото $n = 9$ е малко, тъй че можем да направим сметките на ръка или с електронна таблица.

n	1	2	3	4	5	6	7	8	9
$do(n)$	1	2	6	16	44	120	328	896	2448
$re(n)$	1	3	8	22	60	164	448	1224	3344
$mi(n)$	1	3	8	22	60	164	448	1224	3344
$fa(n)$	1	2	6	16	44	120	328	896	2448
$sol(n)$	1	3	8	22	60	164	448	1224	3344

В последния стълб са бройките на мелодиите от девет ноти според първата нота: 2448 мелодии започват с *до* и т.н. Има общо $2448 + 3344 + 3344 + 2448 + 3344 = 14928$ мелодии от девет ноти.

СХЕМА ЗА ТОЧКУВАНЕ

Задача 1 носи 34 точки, разпределени по подусловия, както е указано на втората страница от домашното. По-точно:

а), б), е) Признават се само решения, които са верни изцяло.

в) — Ако предложеното кодиране на състоянията не е биекция — 0 точки.
— Ако е биекция, но не е достатъчно кратко и ясно — 2 точки.
— Ако липсва пример или има грешка в примера — отнема се 1 точка.

г) — Ако сметките не са доведени докрай или е получен грешен резултат — 0 точки:
за решаване на задачите от домашните работи има много време,
затова технически грешки са недопустими.
— Ако е получен верен, но недостатъчно обоснован резултат — 0 точки.
— Ако е получен верен и обоснован резултат — дават се $\left\lfloor \frac{12}{n} \right\rfloor$ точки,
където n е броят на случаите, на които е разбито решението.

д) — Ако не е намерено решение или предложеното решение е грешно,
или броят на ходовете не е най-малкият възможен — 0 точки.
— Ако решението е вярно и броят на ходовете е минимален — 12 точки.

Задача 2 носи 33 точки, разпределени по следния начин:

— за построяване на матрицата на свързаност на графа: 5 точки;
— за пресмятане на осмата степен на матрицата: 15 точки (по 5 т. за всяко повдигане на квадрат);
— за събиране на елементите на осмата степен на матрицата (получаване на отговора): 5 точки;
— за подробна обосновка (обясняване на смисъла на отделните числа): 8 точки.

Задача 3 носи 33 точки, разпределени по следния начин:

— за определяне на петте функции: 5 точки;
— за поставени начални условия: 5 точки;
— за съставена система от рекурентни уравнения: 5 точки;
— за прилагане на рекурентните уравнения (попълване на динамичната таблица): 5 точки;
— за събиране на числата от последния стълб на таблицата: 5 точки;
— за подробна обосновка (обясняване на смисъла на отделните числа): 8 точки.