

АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 2

2024-2025

- трећи домаћи задатак -

Опште напомене:

1. Домаћи задатак 3 састоји се од једног програмског проблема. Студенти проблем решавају **самостално**, на програмском језику C или C++.
2. Пре одбране, сви студенти раде тест знања који се ради на рачунару коришћењем система *Moodle* (<http://elearning.rcub.bg.ac.rs/moodle/>). **Сви студенти треба да креирају налог и пријаве се на курс пре почетка лабораторијских вежби.** Пријава на курс ће бити прихваћена и важећа само уколико се студент региструје путем свог налога електронске поште на серверу **mail.student.etf.bg.ac.rs**.
3. Реализовани програм треба да комуницира са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
4. Унос података треба омогућити било путем читања са стандардног улаза, било путем читања из датотеке.
5. Решења треба да буду отпорна на грешке и треба да кориснику пруже јасно обавештење у случају детекције грешке.
6. Приликом оцењивања, биће узето у обзир рационално коришћење ресурса. **Примена рекурзије се неће признати као успешно решење проблема које може освојити максималан број поена.**
7. Дозвољено је коришћење линеарних структура података из стандардне библиотеке *stl*.
8. За све недовољно јасне захтеве у задатку, студенти треба да усвоје разумну претпоставку у вези реализације програма. Приликом одбране, демонстраторе треба обавестити које претпоставке и која ограничења су усвојена (на пример, максимална димензија низа и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
9. Одбрана трећег домаћег задатка ће се обавити према распореду који ће накнадно бити објављен на сајту предмета.
10. За решавање задатака који имају више комбинација користити следеће формуле.
(**R** – редни број индекса, **G** – последње две цифре године уписа):

$$i = (R + G) \bmod 2$$

$$j = (R + G) \bmod 3$$

11. Предаја домаћих ће бити омогућена преко *Moodle* система. Детаљније информације ће бити благовремено објављене. Име архиве која се предаје мора бити **piggbbb.zip**. Погрешно назван и предат домаћи задатак повлачи одузимање 25 поена. Закасна предаја домаћег задатка неће бити омогућена. Није потребно предавати материјале са сајта предмета приложене уз домаћи задатак.
12. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака, као и да пријаве теже случајеве повреде Правилника о дисциплинској одговорности студената Универзитета у Београду Дисциплинској комисији Факултета. Током израде решења није дозвољена употреба алата вештачке интелигенције заснованих на великим језичким моделима (*ChatGPT*, *Github Copilot* и сл.).

Задатак 1 – Употреба хипова у имплементацији компјутерске игре [100 поена]

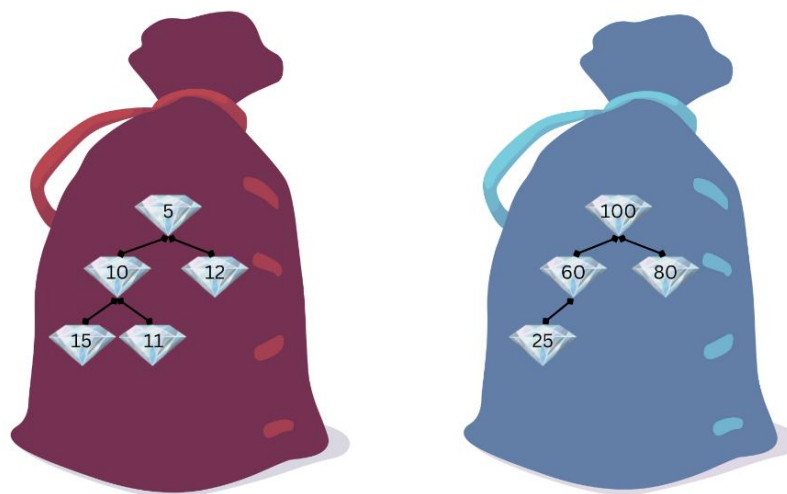
Посматра се авантуристичка компјутерска игра „Пут кроз краљевство Аспире“. Ова игра је игра за једног играча. Играч има улогу ратника који истражује краљевство Аспире, интерагује са окружењем и бори се против непријатеља у циљу очувања краљевства. Потребно је имплементирати мали део ове игре који користи хип структуру података за имплементацију инвентара играча.

Играч у оквиру свог инвентара поседује плаву и црвену врећу. Лутањем кроз напуштена села краљевства наилази на драгуље, дијаманте који су веома вредни и мојсаните који личе на дијаманте, али им је вредност далеко мања. Сваки драгуљ у игри има позитивну целобројну вредност која одређује његову драгоценост. Играч дијаманте смешта у плаву, а мојсаните у црвену врећу. На свом путу, играч понекад, наиђе на путујуће трговце оружја и пекаре. Како му је храна потребна да би одржао енергију, а оружје како би се изборио са непријатељима, сваки пут када наиђе на трговца или пекара, он мења своје драгуље за одговарајућу робу. Путујући пекари не разликују добро драгуље, тако да играч како би их уштедео, сваки пут када се мења са пекарком даје мојсанит са најмањом вредношћу. Међутим, путујући трговци су много искуснији и траже само најдрагоценије дијаманте у замену за оружје, те играч сваки пут када се мења са трговцем даје дијамант са највећом вредношћу. Како би играч у сваком тренутку ефикасно могао да приступи највреднијем дијаманту и најмање вредном мојсаниту, вреће које он има су организоване у виду хипова (врећа са дијамантима у виду максималног хипа, а врећа са мојсанитима у виду минималног хипа). У хиповима је потребно чувати само вредности драгуља.

У зависности од редног броја i проблема који се решава, реализовати:

0. Биномни хип
1. Фибоначијев хип

Током свог пута, играч пролази кроз граничне прелазе између области краљевства где му војници ради контроле траже да покаже шта се налази у његовим врећама. Повремено, играч наиђе и на лопова, када одлучи да све драгуље пребаци у једну врећу и то у ону која је иницијално имала више драгуља (у случају да обе вреће имају исти број драгуља, све пребацује у плаву врећу), како би лопову показао празну врећу и избегао крађу.



Слика 1. Пример садржаја врећа имплементираних помоћу бинарног хипа

Потребно је подржати следеће функционалности игре, које се своде на операције за рад са хиповима:

1. **[5 поена]** Куповина плаве и црвене вреће за играча (стварање празних хипова).
2. **[5 поена]** Поклањање плаве и црвене вреће сиромашнима (уништавање хипова).
3. **[10 поена]** Смештање драгуља у одговарајућу врећу (додавање елемента у максимални или минимални хип).
4. **[10 поена]** Дохватање драгуља из одговарајуће вреће (уклањање максималног елемента из максималног хипа или минималног елемента из минималног хипа).
5. **[25 поена]** Пребацивање свих драгуља у једну врећу (спајање хипова).
6. **[10 поена]** Претрес на граници (испис хипова).

У зависности од редног броја j проблема који се решава, додатно реализовати **[10 поена]**:

0. Наилазак на ретког путујућег трговца оружја ком се даје k највреднијих дијаманата (уклањање k највреднијих дијаманата из вреће).
1. Наилазак на ретког путујућег пекара ком се даје k најмање вредних мојсанита (уклањање k најмање вредних мојсанита из вреће).
2. Наилазак на ретког мешовитог трговца ком се даје $k/2$ највреднијих дијаманата и $k/2$ најмање вредних мојсанита.

[25 поена] Потребно је подржати и симулацију игре, која подразумева учитавање низа команди, односно догађаја из текстуалне датотеке и њихово сукцесивно извршавање. У текстуалној датотеци, једна команда је задата у једном реду једнословном ознаком и евентуалним додатним податком у угластим заградама. Опис команди се налази у наставку:

- V – Куповина плаве и црвене вреће.
- B – Поклањање плаве и црвене вреће.
- D[x] – Проналажење дијаманта чија је вредност x .
- M[x] – Проналажење мојсанита чија је вредност x .
- T – Наилазак на путујућег трговца оружја.
- P – Наилазак на путујућег пекара.
- G – Наилазак на гранични прелаз.
- L – Наилазак на лопова.
- R[k] – Наилазак на ретког трговца који захтева k драгуља.

Корисник са програмом интерагује путем једноставног менија. Програм треба да испише садржај менија, а затим да чека да корисник изабере (унесе путем тастатуре) редни број неке од понуђених ставки, након чега, пре извршења, од корисника очекује да по потреби унесе додатне параметре. Поступак се понавља све док корисник у менију не изабере опцију за прекид програма.

Напомене

По потреби реализовати и додатне методе, где је то примерено.

За тестирање програма се могу користити датотеке које се налазе у оквиру посебне архиве.

Рад са датотекама у језику C++ захтева увожење заглавља `fstream` (именски простор `std`). За читање података користи се класа `ifstream`. Након отварања датотеке, читање се врши на исти начин као и са стандардног улаза. Кратак преглед најбитнијих метода и пријатељских функција ове класе је дат у наставку.

<pre>void open(const char *_Filename, ios_base::openmode _Mode = ios_base::in, int _Prot = (int)ios_base::_Openprot);</pre>	Отвара датотеку задатог имена за читање. <code>ifstream dat;</code> <code>dat.open("datoteka.txt");</code>
<pre>void close();</pre>	Затвара датотеку.
<pre>bool is_open();</pre>	Утврђује да ли је датотека отворена.
<pre>operator>></pre>	Преклопљен оператор за просте типове података.
<pre>ifstream dat; dat.open("datoteka.dat"); if(! dat.is_open()) greska(); char niz[20]; dat >> niz; dat.close();</pre>	Пример отварања датотеке, провере да ли је отварање успешно, читање једног знаковног низа из датотеке и затварања датотеке.