

Пројектовање и анализа алгоритама

Трећа лабораторијска вежба

Задатак 0.

Имплементирати алгоритам који је основа ZIP-а (по алгоритму представљеном на настави).

За тестирање кодирати случајно генерисане низове карактера од 100, 1000, 10к, 100к, 1М и 10М елемената и "Lorem Ipsum" (<https://www.lipsum.com/>) текст надовезан више пута док се не добије тражена дужина низа.

Низ карактера треба учитати из фајла, кодирати и резултат са пратећим речником (за алгоритме са префиксним кодовима) уписати у фајл.

Потврдити исправност кодирања тако што ће ваш програм учитати и декодирати кодирани фајл. Декодирану вредност уписати у нови фајл.

Као резултат задатка послати једну архиву која садржи изворни код, као и фајл са резултатима кодирања и проценом степена остварене компресије.

Степен компресије се рачуна као:

$100 * (1 - (\text{величина компресованог фајла укључујући и речник}) / (\text{величина полазног фајла}))$

Задатак 1.

Користећи приступ похлепног алгоритма, наћи најмањи производ који чине елементи континуираног подниза датог низа. Континуирани подниз може да буде сачињен и од само једног елемента, а може да буде и цео низ.

Примери:

Улаз: $a[] = \{-1, -1, -2, 4, 3\}$

Излаз : -24

Објашњење : Минимални производ ће бити $(-2) * (-1) * (-1) * 4 * 3 = -24$

Улаз: $a[] = \{-1, 0\}$

Излаз : -1

Објашњење: -1 (један елемент) је минимални могући производ

Улаз: $a[] = \{0, 0, 0\}$

Излаз: 0

Задатак 2.

Нека је дато N мишева који треба да заузму N рупа које су једна поред друге. На роџетки знамо позиције мишева и позиције рупа. Миш може да остане на својој позицији, да се помери за један корак удесно од k до $k + 1$ или да се помери за један корак лево од k до $k - 1$ места. Сваки од ових потеза траје 1 минут. Распоредите мишеве тако да време када последњи миш уђе у рупу буде минимално.

Улаз: позиције мишева су:

4 -4 2

Положаји рупа су:

4 0 5

Излаз: 4

Нека миш на позицији 4 иде у рупу на позицији 4. Потребно време је 0 минута

Нека миш на позицији -4 иде у рупу на позицији 0. Потребно време је 4 минута

Нека миш на позицији 2 иде у рупу на позицији 5. Потребно време је 3 минута

После 4 минута сви мишеви су у рупама.

Улаз: позиције мишева су:

-10, -79, -79, 67, 93, -85, -28, -94

позиције рупа су:

-2, 9, 69, 25, -31, 23, 50, 78

Излаз : 102

Задатак 3.

Дато је N активности са својим временом почетка и завршетка. Изаберите максималан број активности које може да обави једна особа, под претпоставком да особа може истовремено да ради само на једној активности. Приказати индексе тих активности.

Улаз: почетак[] = {10, 12, 20}, крај[] = {20, 25, 30}

Излаз: активност 0

Објашњење: Особа може обављати највише једну активност.

Улаз: почетак[] = {1, 3, 0, 5, 8, 5}, завршетак[] = {2, 4, 6, 7, 9, 9};

Излаз: активности 0 1 3 4

Задатак 4

У оперативним системима, кад год је нова страница захтевана и није присутна у меморији, генерише се грешка странице и оперативни систем замењује једну од постојећих страница новом потребном страницом. Различити алгоритми замене страница предлажу различите начине да одлучите коју страницу да замените. Циљ свих алгоритама је смањење броја грешака страница. У овом алгоритму, ОС замењује страницу која се неће користити најдужи временски период у будућности.

Ако је референтна страница већ присутна, повећајте број погодака.

Ако није присутна, пронађите страницу која се никада неће спомињати у будућности. Ако таква страница постоји, замените ову страницу новом. Ако таква страница не постоји, пронађите страницу која је најдаље референцирана у будућности. Замените ову страницу новом.

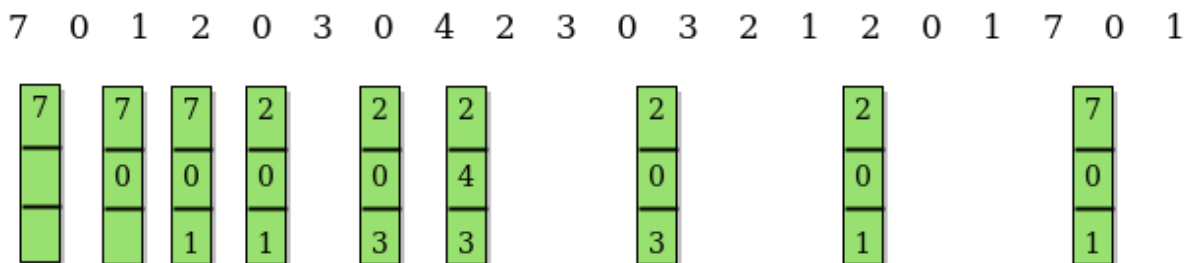
Одредити укупан број погодака и промашаја.

Улаз: Број фрејмова у кешу, $fn = 3$

Референтни низ страница, $pg[] = \{7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1\}$;

Израз: Број погодака = 11

Број промашаја = 9



Улаз: Број фрејмова у кешу, $fn = 4$

Референтни низ страница, $pg[] = \{7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2\}$;

Израз: Број погодака = 7

Број промашаја = 6

Задатак 5

Дат је низ целих бројева где сваки елемент представља максималан број корака који се могу направити напред од тог елемента. Напишите функцију за враћање минималног броја скокова до краја низа (почевши од првог елемента). Ако је елемент 0, онда се не можемо кретати кроз тај елемент. Вратити -1 ако није могуће доћи до краја.

Улаз: arr[] = {1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9}

Излаз: 3 (1-> 3 -> 8 -> 9)

Објашњење: Скочите са 1. елемента на 2. елемент пошто постоји само 1 корак, сада постоје три опције 5, 8 или 9. Ако је изабрано 8 или 9 онда се може доћи до крајњег чвора 9. Тако су направљена 3 скока.

Улаз: arr[] = {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}

Излаз: 10

Објашњење: У сваком кораку потребан је скок тако да је број скокова 10.

Задатак 6

На берзи постоји производ са својих бесконачно акција. Цене акција су дате за H дана, где $цена[u]$ означава цену акције на u -ти дан. Постоји правило да купац може да купи највише u акција на u -ти дан. Ако купац у почетку има k динара, одредите максималан број акција које купац може купити.

На пример, за 3 дана цена акције је дата као 7, 10, 4. Можете купити 1 акције у вредности од 7 РСД на дан 1, 2 акције у вредности од по 10 РСД на дан 2 и 3 акције у вредности од по 4 РСД на дан 3.

Улаз : цена [] = { 10, 7, 19 }, $k = 45$.

Излаз: 4

Објашњење: Купац купује 1 акцију првог дана за 10 динара, 2 акције 2. дана по 7 РСД и 1 акцију 3. дана за 19 РСД.

Дакле, укупно 10, $7 * 2 = 14$ и 19 респективно. Укупан износ је $10 + 14 + 19 = 43$ и број купљених акција је 4.

Улаз: цена[] = { 7, 10, 4 },

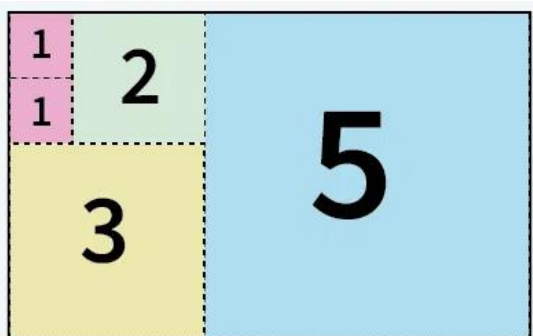
$k = 100$.

Излаз: 6

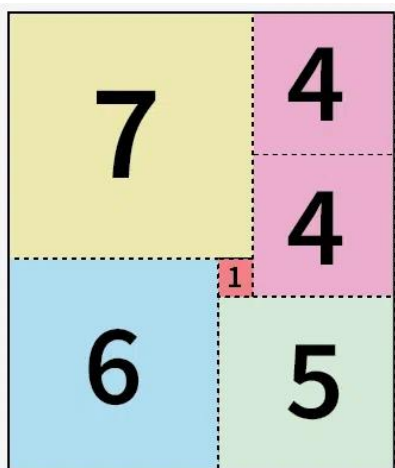
Задатак 7

Дат је папир величине $A \times B$, задатак је да се цео папир исече на квадрате било које величине. Пронађите минимални број квадрата који се могу исећи из папира.

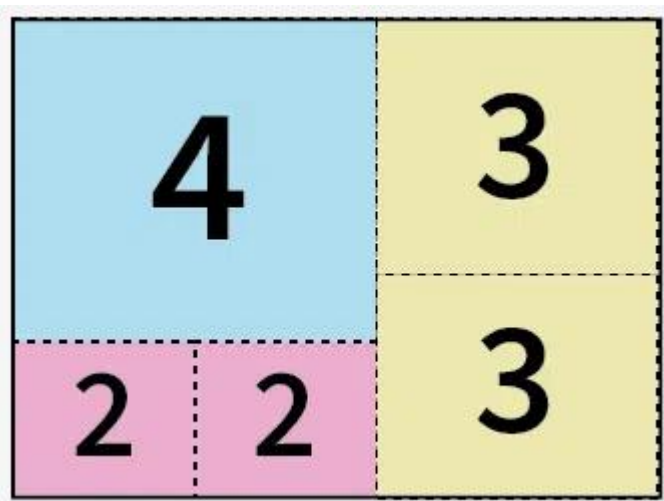
Улаз: $A = 5$, $B = 8$



Улаз: $A = 13$, $B = 11$



Улаз: $A = 6$, $B = 7$



Задатак 8

Имплементирати Хафманов алгоритам за кодирање по поступку описаном на настави.

За тестирање кодирати случајно генерисане низове карактера од 100, 1000, 10к, 100к, 1М и 10М елемената, и "Lorem Ipsum" (<https://www.lipsum.com/>) текст надовезан више пута док се не добије тражена дужина низа.

Низ карактера треба учитати из фајла, кодирати и резултат са пратећим речником (за алгоритме са префиксним кодовима) уписати у фајл.

Потврдити исправност кодирања тако што ће ваш програм учитати и декодирати кодирани фајл. Декодирану вредност уписати у нови фајл.

Као резултат задатка послати једну архиву која садржи изворни код, као и фајл са резултатима кодирања и проценом степена остварене компресије.

Степен компресије се рачуна као:

$100 * (1 - (\text{величина компресованог фајла укључујући и речник}) / (\text{величина улазног фајла}))$

Задатак 9

Имплементирати Шанон-Фано алгоритам за кодирање.

За тестирање кодирати случајно генерисане низове карактера од 100, 1000, 10к, 100к, 1М и 10М елемената, и "Lorem Ipsum" (<https://www.lipsum.com/>) текст надовезан више пута док се не добије тражена дужина низа.

Низ карактера треба учитати из фајла, кодирати и резултат са пратећим речником (за алгоритме са префиксним кодовима) уписати у фајл.

Потврдити исправност кодирања тако што ће ваш програм учитати и декодирати кодирани фајл. Декодирану вредност уписати у нови фајл.

Као резултат задатка послати једну архиву која садржи изворни код, као и фајл са резултатима кодирања и проценом степена остварене компресије.

Степен компресије се рачуна као:

$100 * (1 - (\text{величина компресованог фајла укључујући и речник}) / (\text{величина полазног фајла}))$