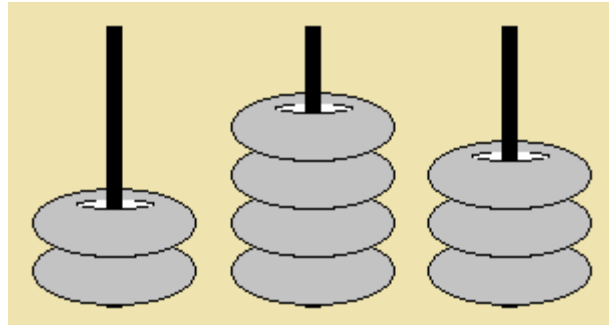




Домаћи задатак из предмета Интелигентни системи за јануарски рок школске 2017/18. године

Основне информације

Ним (енг. *nim*) је логичка игра која се игра између два играча који наизменично повлаче потезе. На почетку игре су жетони распоређени по стубовима. При сваком потезу, играч који повлачи потез узима жетоне (бар један), само са једног стуба, тако да ни у једном тренутку у току игре не постоје два стуба са истим бројем жетона (осим 0). Победник је онај играч који узме последњи жетон, или онај играч који није дошао у ситуацију да не може да одигра потез.



Правила игре (модификовани) Ним

- Постоји n стубова са жетонима. Пре почетка игре на сваки стуб се поставља до m жетона. На разним стубовима се мора налазити различит број жетона. На сваком стубу мора бити барем један жетон. Мора постојати дозвољен потез за првог играча на почетку игре.
- Игру играју два играча, који повлаче потезе наизменично. Потез играча је узимање одређеног броја жетона (барем једног), само са једног стуба. Број узетих жетона у једном потезу не сме бити већи од двоструког броја жетона узетих у претходном потезу другог играча. Играч може да узме жељени број жетона само уколико то узимање неће проузроковати нерегуларну ситуацију у којој ће постојати два стуба са истим бројем жетона (осим 0).
- Игра се може завршити на два начина:
 - када ни на једном стубу више нема жетона – победник је играч који је узео последњи жетон;
 - када играч који је на потезу не може да одигра потез јер би сваким проузроковао нерегуларну ситуацију у којој би постојала два стуба са истим бројем жетона (осим 0) – победник је играч који није на потезу.



Кориснички захтеви

- Изглед корисничког интерфејса

У сваком тренутку треба да буду приказани стубови са жетонима и информација који играч је на потезу.

- Покретање игре

Пре почетка игре неопходно је:

- одабрати ко игра игру (човек против човека, човек против рачунара или рачунар против рачунара),
- одабрати број стубова (највише 10) и почетни распоред жетона по стубовима (највише 10 жетона на стубу),
- одабрати тежину игре – дубину развијања стабла игре, у случају да је бар један од играча рачунар.

- Решавање проблема

Проблем је потребно решити *MinMax* алгоритмом.

Потребно је имплементирати једноставног играча. Овај играч треба да користи једноставну статичку функцију процене f , користећи само број жетона на стубовима $f = m_1 \oplus m_2 \oplus \dots \oplus m_n$, где је m_i број жетона на стубу i .

Након креирање једноставног играча, потребно је имплементирати једноставног алфа-бета играча, који побољшава једноставног играча тако што приликом претраживања користи алфа-бета одсецање.

Након креирања једноставног алфа-бета играча, потребно је направити такмичарског алфа-бета играча, који треба да оптимизује перформансе претходног. Можете укључити било које технике које желите (изузев нити, GPU програмирања или других сличних ствари) за реализацију овог играча. Почните од надоградње постојећег алфа-бета играча, коришћењем боље статичке функције процене, која ће користити више информација из стања игре.

- Имплементација игре

Препоручено је да се имплементација ради у програмском језику Јава или C++/C#, у развојном окружењу *Eclipse IDE* или *VisualStudio*, уз дозвољено коришћење свих стандардних библиотека и структура података. Дозвољено је имплементирати систем и у неком другом програмском језику по жељи (Android, Swift for iOS, ...).



Главни програм, алгоритме, основне структуре и модове рада симулације, одвојити у засебне класе. Сваки студент треба да направи свој пакет (или *namespace*) `etf.nim.<username>` где је `<username>` Ваше корисничко име за портал е-Студент (у формату *piGGBBBBx*, где су *pi* иницијали - презиме и име, ознака *GG* представља последње две цифре године уписа факултета, ознака *BBBB* представља четвороцифрени број индекса, проширен водећим нулама, а ознака *x* представља ниво студија *d* или *m*). Студенти су обавезни да у том пакету имплементирају комплетан код, а не ван њега.

- Корисници интерфејс треба да буде интуитиван и довољно детаљан приликом решавања проблема - не смеју се прескакати кораци алгоритма.

Захтеви домаћег задатка

Ваш задатак је да реализуете игру Ним коришћењем теорије игара. За максималан број поена, програм мора да испуни све услове пројектног задатка и да ради поуздано.

Поред тога, треба написати извештај, који описује како је проблем решен. Извештај треба да садржи:

- Кратак приказ имплементираног алгоритма
- Кратке описе свих реализованих класа, са потписима и описима метода

Извештај приложити уз код, у фолдеру `/documentation` и именовати га као: `Prezime_Ime_IS_Domaci_Jan2018.pdf`.

Напомене

Електронску верзију решења овог домаћег задатка предати најкасније до 12. јануара 2018. године у 23:59, као ZIP архиву, на [сајту](#) за предају домаћег. Термин одбране домаћег задатка биће накнадно објављен.

Домаћи задатак из предмета *Интелигентни системи* се ради самостално и није обавезан за полагање испита (на испиту се може заменити теоријским питањима из целокупног градива). Овај домаћи задатак се може бранити **само у јануарском испитном року**. Домаћи задатак вреди максимално до 20 поена (уз могућност додатних поена за радове који се посебно истакну).

На усменој обрани кандидат мора самостално да покрене своје решење. Кандидат мора да поседује потребан ниво знања о задатку, мора да буде свестан недостатака приложеног решења и могућности да те недостатке реши. Кандидат мора тачно да одговори и на одређен број питања која се баве тематиком домаћег задатка.

Пре започињања реализације проблема или тражења помоћи задатак прочитати у целини и пажљиво. Уколико у задатку нешто није довољно прецизно дефинисано, од кандидата се очекује да уведу разумне претпоставке и темељно их образложе.