

Основне академске студије
Информациони инжењеринг

Основи рачунарске интелигенције

Навођене претраге

(материјали за вежбе)

1. Увод
2. Похлепна претрага
3. Претрага A^*
4. Претрага IDA^*
5. Примена претрага

Навођене претраге

напреднија класа претрага у односу на ненавођене претраге
општи ток претраге сличан оном код ненавођених претрага
постоји разлика у начину избора чвора графа за испитивање
користе се додатне информације о задатку током трагања за
решењем

додатне информације нису директно уграђене у опис задатка
додатне информације обично се тичу домена којем припада задатак
додатне информације служе као помоћ приликом претраге

Навођене претраге

хеуристичка функција h

вредност хеуристичке функције за посматрани чвор графа представља процену цене најповољније путање од посматраног до одредишног чвора

вредности функције су ненегативне

вредност нула означава да је достигнут одредишни чвор

додатне информације о задатку су представљене преко хеуристичке функције

функција евалуације f

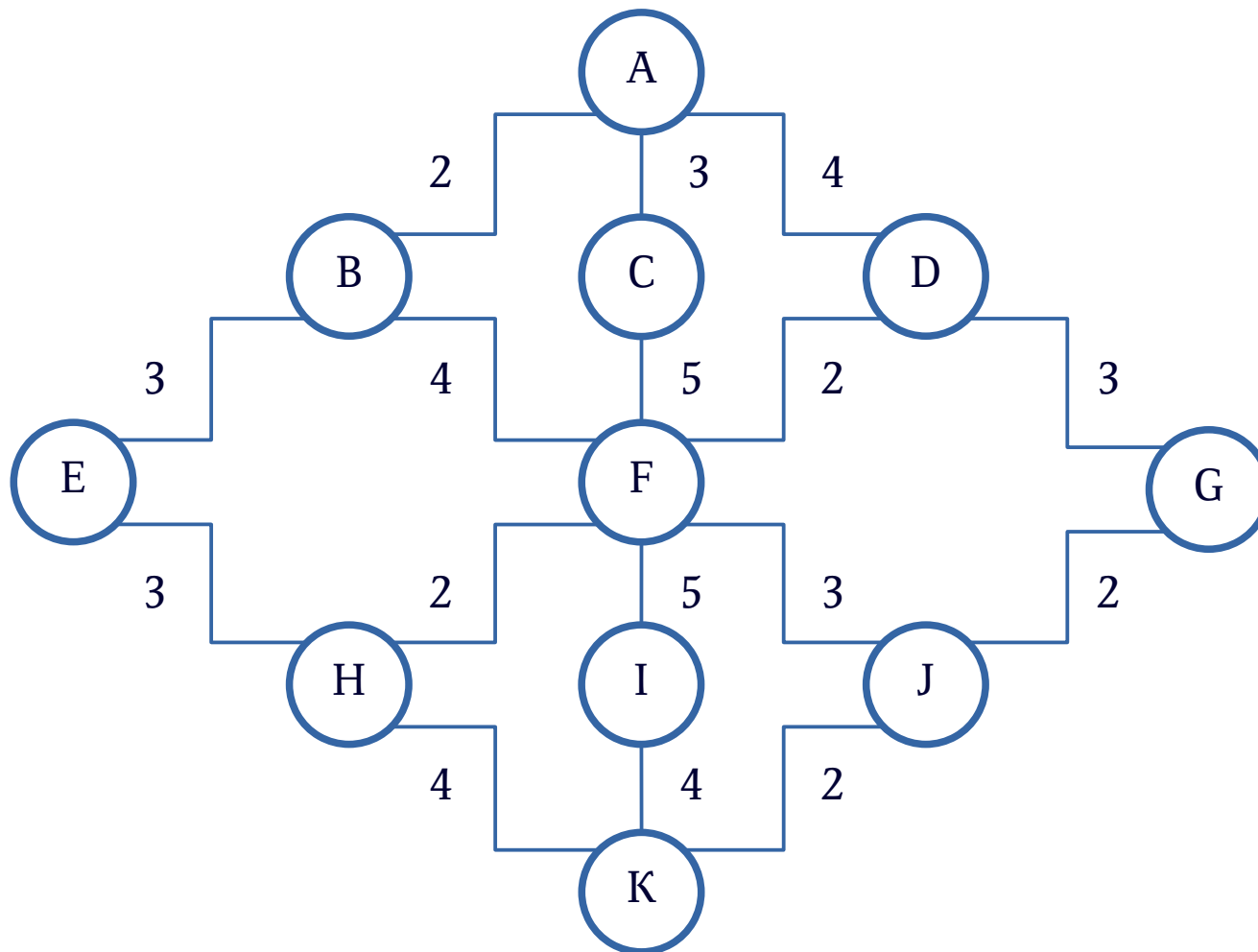
вредност функције евалуације за посматрани чвор графа описује колико је чвор погодан да постане део путање по којој се спроводи претрага

током претраге испитује се онај чвор за који се добија најбоља вредност функције евалуације

у случају цена, најмања вредност је најбоља

хеуристичка функција се уграђује у функцију евалуације

Граф X



1. Увод
- 2. Похлепна претрага**
3. Претрага A^*
4. Претрага IDA^*
5. Примена претрага

Похлепна претрага

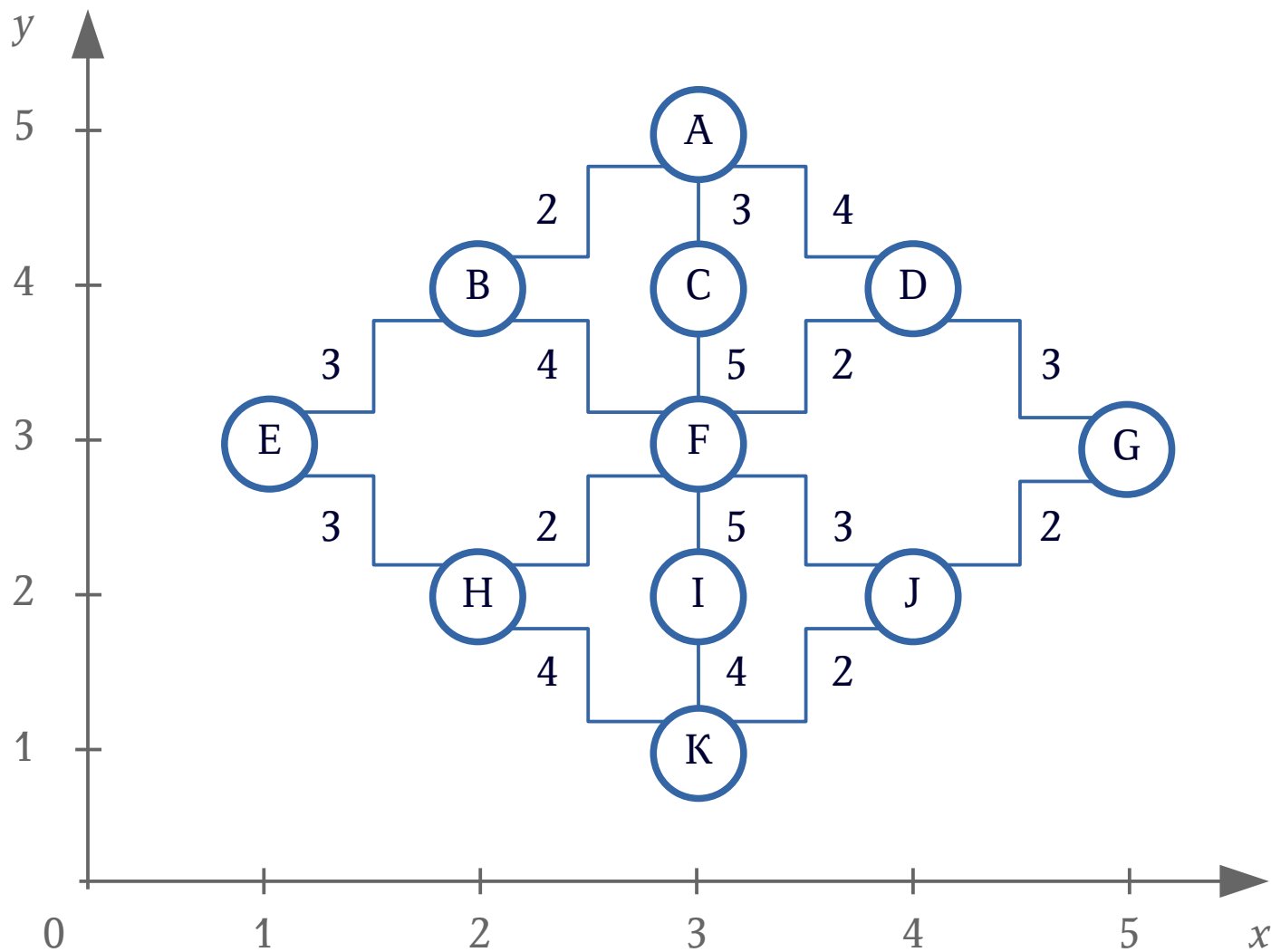
током претраге бирају се они чворови графа за које се процењује да на нивоу тренутне итерације на најповољнији начин воде до одредишног чвора

користи се функција евалуације која се своди на хеуристичку функцију у свакој итерацији испитује се онај чвор за који хеуристичка функција има најмању вредност

решење не мора бити пронађено, чак и када постоји пронађено решење не мора бити оптимално

Похлепна претрага

Граф X



Похлепна претрага

Задатак 1.

Имплементирати класу **GreedySearch** која подржава похлепну претрагу над графом. Приликом позива конструктора проследити граф, ознаку полазног чвора, ознаку одредишног чвора и хеуристичку функцију.

Покренути претрагу графа X са чвором A као полазним и чвором K као одредишним. Као хеуристику користити Менхетн удаљеност од одредишног чвора.

1. Увод
2. Похлепна претрага
- 3. Претрага A^***
4. Претрага IDA^*
5. Примена претрага

Претрага A^*

током претраге бирају се они чворови графа који су најповољнији у погледу и већ пређеног пута од полазног чвора и процењеног предстојећег пута до одредишног чвора

користи се функција евалуације која се заснива на збиру вредности две функције

- функција која за посматрани чвор даје цену пређеног пута од полазног до посматраног чвора

- хеуристичка функција

Задатак 2.

Имплементирати класу **AStarSearch** која подржава претрагу A^* над графом. Приликом позива конструктора проследити граф, ознаку полазног чвора, ознаку одредишног чвора и хеуристичку функцију.

Класа **GreedySearch** може послужити као полазиште за имплементацију.

Покренути претрагу графа X са чвором A као полазним и чвором K као одредишним. Као хеуристику користити Менхетн удаљеност од одредишног чвора.

Упоредити добијено решење с оним које даје претрага заснована на класи **GreedySearch**.

1. Увод
2. Похлепна претрага
3. Претрага A^*
- 4. Претрага IDA^***
5. Примена претрага

Претрага *IDA**

претрага се спроводи итеративно за задате граничне вредности функције евалуације

у сваком циклусу постоји гранична вредност

претрага креће од полазног чвора графа и испитују се само они чворови чија је вредност функције евалуације мања или једнака са граничном вредношћу

уколико је неки чвор испитан заједно са свим својим следбеницима, онда може бити уклоњен из меморије заједно са следбеницима

у првом циклусу гранична вредност одговара вредности функције евалуације за полазни чвор

у наредним циклусима гранична вредност одговара најмањој међу вредностима функције евалуације које су у претходној итерацији биле изнад тадашње граничне вредности

Задатак 3.

Имплементирати класу **IDASearch** која подржава претрагу IDA^* над графом. Приликом позива конструктора проследити граф, ознаку полазног чвора, ознаку одредишног чвора и хеуристичку функцију.

Класа **AStarSearch** може послужити као полазиште за имплементацију.

Покренути претрагу графа X са чвором A као полазним и чвором K као одредишним. Као хеуристику користити еуклидску удаљеност од одредишног чвора.

1. Увод
2. Похлепна претрага
3. Претрага A^*
4. Претрага IDA^*
- 5. Примена претрага**

Задатак 4.

Имплементирати класу **TripSearch** која подржава тражење пута између датих места применом похлепне претраге. Приликом позива конструктора проследити листу места, матрицу путева, назив полазног места, назив циљног места и хеуристичку функцију.

Листа места садржи називе n места. Матрица путева је димензија $n \times n$ и садржи дужине путева који повезују два места без трећих места између.

Вредност на координатама i, j у матрици је дужина пута који директно повезује i -то место из листе с j -тим местом из листе.

Вредност -1 на координатама i, j у матрици означава да не постоји директна веза између одговарајућих места.

Покренути претрагу, а као хеуристику користити удаљеност ваздушном линијом од циљног места.