

Postavljanje hiš in podjetji v ulici dolžine n

Računalništvo 1

Avtor naloge:

Gal Zakrajšek

Ljubljana, 2020

Contents

[Problem 3](#_Toc30785045)

[Primeri 3](#_Toc30785046)

[Ideja rešitve 4](#_Toc30785047)

[Dodajanje vrste z dvema hišama 5](#_Toc30785048)

[Dodajanje vrste z hišo in podjetjem 5](#_Toc30785049)

[Izboljšave algoritma 6](#_Toc30785050)

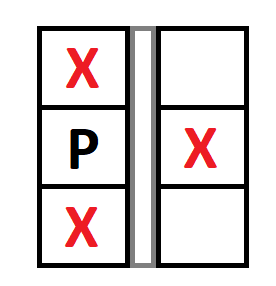
[Memoizacija in časovna zahtevnost 6](#_Toc30785051)

[Prostorska zahtevnost 6](#_Toc30785052)

[Viri 7](#_Toc30785053)

# Problem

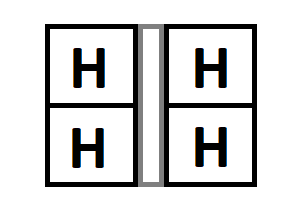
Dano imamo dolžino ulice , ki nam pove, da imamo na voljo parcel, na katere bomo postavljali hiše ali pa podjetja. Pri postavljanju morajo biti vse parcele vedno zasedene, imamo pa tudi dodatne zahteve in sicer, da nobeno podjetje ni sosednjo z drugim podjetjem. Prav tako podjetje ne sme stati nasproti drugega podjetja. Medtem ko hiše lahko postavljamo na katero koli parcelo. Naš cilj bo napisati funkcijo, ki nam bo vrnila število vseh možnih načinov, da postavimo hiše in podjetja, da bodo zgornje zahteve izpolnjene. V slikah bomo označevali podjetja z črko P in hiše z črko H. Naslednja slika prikazuje zgornja pravila in sicer so z X označene parcele, kjer se podjetja ne smejo prikazati, saj bi bili sosednji označenemu podjetju.



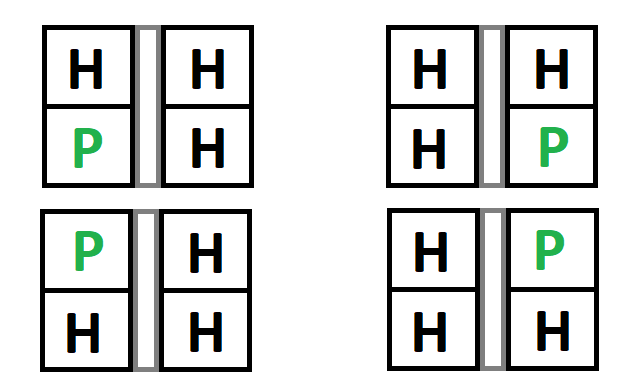
# Primeri

Za lažjo predstavo, si bomo ogledali primer za , saj je to še edini primeren pri katerem lahko narišemo vse možnosti, kajti že pri imamo 17 različnih postavitev.

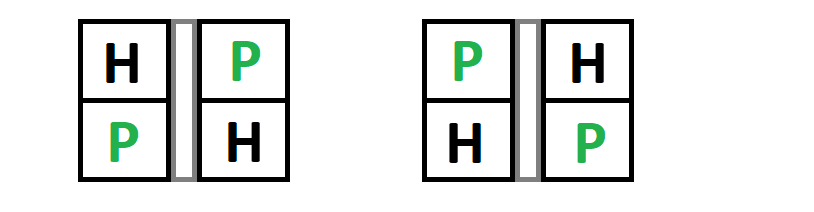
Začnimo kar z tem, da lahko vedno vse parcele zasedejo hiše, saj nimamo nobenih pravil, ki bi to omejevale.



Potem lahko v našo postavitev postavimo eno podjetje, tukaj imamo 4 različna mesta na katera ga lahko postavimo.



Ker se v isti vrsti nikoli ne smeta pojaviti dve podjetji vemo, da je lahko v ulici dolžine največ postavljenih podjetji. Zato ugotovimo, da imamo lahko ob cesti dolžine največ 2 podjetji. Postavimo jih lahko na sledeča dva načina:



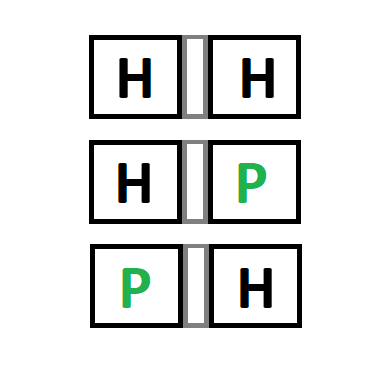
Iz tega primera je razvidno, da imamo 7 različnih načinov postavitve naših hiš in podjetji v ulico dolžine .

# Ideja rešitve

Ena ideja rešitve bi lahko bila, da najprej ustvarimo vse možne načine postavitve hiš in podjetji na razpoložljive parcele in nato preverimo vsak način posebej, če le ta ustreza vsem zahtevam. Pri taki ideji bi hitro prišli do težave in sicer teh načinov, bi bilo hitro zelo veliko. Za primer lahko pogledamo ulico z dolžino 4, kar bi pomenilo, da imamo na voljo 8 parcel. V tem primeru bi bilo vseh možnih načinov . V še večjih primerih število eksplodira in bi za preverjanje ustreznosti vseh načinov porabili ogromno časa. Zato ta ideja ni najboljša.

Če malo razmislimo in se reševanja lotimo drugače lahko pridemo do ugotovitve, da če sprva rešimo naš problem za majhne -je potem lahko manjše probleme razširimo in tako pridemo do rešitve večjih -jev.

Sprva ugotovimo, da če primer razširimo iz na bomo v naši ulici dodali eno vrsto in sicer dve parceli vsako na eni strani ceste. Ti dve parceli lahko zapolnimo na tri različne načine:



1. Dodamo dve hiši
2. Dodamo hišo in podjetje
3. Dodamo podjetje in hišo

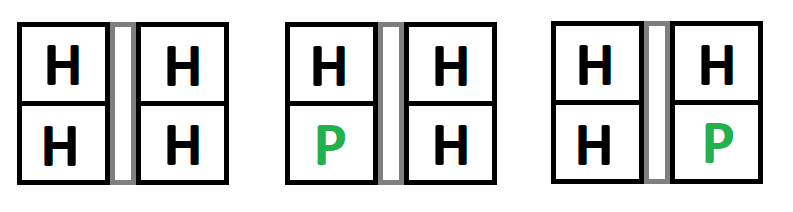
Sedaj se posvetimo ugotavljanju, na katere načine, ki smo jih dobili z dolgo cesto lahko dodamo novo vrsto, ki je med zgoraj naštetimi. Če preštejemo te razširitve za vse tri možne postavitve in jih med sabo seštejemo, bomo dobili ravno vse možne načine za dolgo cesto.

## Dodajanje vrste z dvema hišama

Poglejmo si, katere načine lahko razširimo z vrsto, ki vsebuje dve hiši. Ta del je preprost, saj nobena zahteva ne zavzema postavljanja hiš, kar pomeni, da jih lahko postavimo na katerokoli parcelo ne glede na to, kaj je v sosednjih vrstah. Tako ugotovimo, da vrsto z dvema hišama lahko postavimo na katerikoli način, ki smo ga dobili iz dolge ceste in zato je teh načinov ravno rezultat naše funkcije za dolžino ulice . Za lažjo predstavo, imamo sedaj sliko ki prikazuje sprva vse možne postavitve za ulico dolžine 1.



Sedaj, si predstavljamo, da razširimo zgornje vrste in sicer dodamo eno vrsto, ki ima dve hiši. Hitro ugotovimo, da lahko to postavimo na vse tri zgornje ulice in dobimo te tri rešitve.



## Dodajanje vrste z hišo in podjetjem

Izračunati potrebujemo še število načinov, na katere lahko postavimo vrsto, z eno hišo in enim podjetjem. Ker imamo pravilo, da dve podjetji ne smeta biti sosednji, moramo ugotoviti, na koliko načinov se rešitve pri cesti z dolžino končajo z enim podjetjem na vrhu. Ta mora biti na drugi strani kakor podjetje v vrsti, ki jo dodajamo. Število bomo dobili s pomočjo formule, ki jo bomo izpeljali.

Najprej pogledamo število ugodnih razporeditev, ki se končajo z eno pisarno. To število najdemo tako, da vzamemo vse razporeditve za cesto in od nje odštejemo vse razporeditve, ki so se končale z dvema hišama, to smo v prejšnjem razdelku že ugotovili, da se izračuna oziroma dobi tako, da pogledamo rešitve za cesto dolžine . Zdaj imamo število, ki nam pove koliko razporeditev se je končalo z eno hišo in enim podjetjem, ampak sedaj je podjetje lahko na levem ali na desnem delu ulice. Rešitev, z podjetjem na levem koncu v zadnji vrsti bo vedno enako rešitvam z podjetjem na desnemu, saj velja simetrija, zato število, ki smo ga dobili delimo z dve in pridemo do željenega števila.

Z to formulo, dobimo vse načine, ki se končajo z enim podjetjem v zadnji vrsti, ki je na ravno obratnem mestu kakor podjetje v vrsti, ki jo dodajamo. Če sedaj od vseh načinov, ki jih dobimo za cestno dolžine odštejemo zgoraj pridobljeno formulo, dobimo število vseh načinov na katere lahko dodamo novo vrsto z eno pisarno in podjetjem. Končna formula za izračun tega dela je:

Sedaj imamo vse potrebne formule, da izračunamo končni rezultat, ki ga dobimo tako, da seštejemo vse te možnosti. Pri tem moramo biti pozorni, saj lahko dodamo vrsto z hišo in podjetjem na dva načina, saj je lahko podjetje na levem ali desnem delu vrste. Vendar to naše formule ne spremeni veliko, saj samo formulo z drugega razdelka pomnožimo z 2. Končna formula je sledeča:

Formula je podana na dolg način, da je razumljivo kako pridemo do rešitve. Sedaj lahko formulo tudi malo okrajšamo in dobimo končno formulo:

To je sedaj naša rekurzivna formula, z katero lahko računamo rešitve za katerokoli dolžino ceste.

# Izboljšave algoritma

Sedaj smo prišli, do rekurzivnega algoritma, ki nas pripelje do rešitve. Ker pa poznamo tudi osnove dinamičnega programiranja pa lahko ta algoritem še izboljšamo in sicer prva ideja je memoizacija.

## Memoizacija in časovna zahtevnost

Pri memoizaciji vemo, da jo lahko uporabimo takrat, ko potrebujemo nek podatek večkrat in ga do sedaj vedno računamo ponovno. Tega se lahko znebimo tako, da naša funkcija sproti zapisuje vse rešitve, ki jih izračuna v neko tabelo. Tako poskrbimo, da ko bo algoritem ponovno potreboval neko rešitev, je ne bo računal ponovno ampak jo bo šel iskat v tabelo shranjenih rešitev. Z tem zmanjšamo časovno zahtevnost. Tako naš algoritem sedaj deluje v časovni zahtevnosti .

## Prostorska zahtevnost

Po optimizaciji našega algoritma z memoizacijo imamo sedaj algoritem prostorske zahtevnost , saj imamo tabelo vseh rešitev, ki smo jih izračunali. Če pa ponovno pogledamo formulo za izračun, pa ugotovimo, da vedno potrebujemo samo zadnji dve izračunani rešitvi. Zato bi lahko namesto tabele uporabili dve spremenljivki, v kateri bi si shranjevali rešitvi. Tako se znebimo dodatne tabele in svoj algoritem spravimo na prostorsko zahtevnost .

# Viri

Sharma A. *Count ways to build street under given constraints | GeeksforGeeks (b. d.) Pridobljeno s* <https://www.geeksforgeeks.org/count-ways-build-street-given-constraints/> [23. 1. 2020]