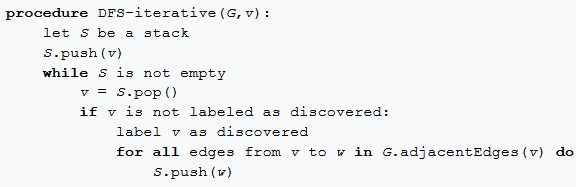
Pregled v globino vizualizacija – Opis razvoja

## Povzetek

Vizualizacija je spisana z uporabo WindowsForms knjižnice v C# jeziku. Za IDE sem uporabil Visual Studio 15. Za izhodišče vizualizacije sem izbral nerekurzivni algoritem iskanja v globino.



Vir: Wikipedia 1

Ker vizualizacijo poganjamo s klikom na gumb Naprej, je bilo treba algoritem prepisati brez zanke while(drugače se koda izvede v trenutku in vizualizacija ni ustrezna).

## KONCEPT – IDEJA

DFS sam po sebi ni zelo zahteven algoritem. Za vizualizacijo pa je treba premisliti na katerih mestih ga ustavimo za čimboljšo uporabniško izkušnjo. Najprej bom opisal kako se izvede vizualizacija z uporabo potrebnih metod znotraj Form1 razreda, nato sledi še opis kode in vseh uporabljenih metod(in orodij). Vmes bom še predstavil diagram poteka.

## 1.Pristop

Ker je najbolj intuitivno uporabiti while zanko tako kot v psevdokodi zgoraj, sem se najprej prikaza lotil z večnitnim programiranjem, kjer se znotraj glavne niti(main thread) vrši koda – vizualizacija, dokler v sklad ne dodamo novega vozlišča. Takrat se glavna nit ustavi in čaka da se izvede pomozna nit(pritisnemo gumb za Naprej). Ko se ta zakljuci se glavni proces nadaljuje naprej, dokler spet ne dodamo vozlišča oz. se algoritem zaključi. Za večnitno programiranje sem uporabil razred AutoRestEvent, ki ga uvozimo iz System.Threading knjižnice. Zaradi nestabilnosti pri uporabi te rešitve sem se na koncu odločil za drugačen pristop, zato se v podrobnosti večnitne rešitve v tem poročilu ne bom spuščal.

## 2.Pristop

Celotna logika algoritma se izvaja znotraj funkcij, ki ju pokličeta gumba Generiraj in Naprej. Na začetku, zunaj konstruktorja definiramo potrebne spremenljivke:

* SlovarSosedov – slovar, ki kot ključe vsebuje zaporedno številko(int) posameznega vozlišča, kot vrednosti pa hrani sezname sosednjih vozlišč. SlovarSosedov se naključno ustvari ob kliku na gumb Generiraj.
* Seznam Point objektov – Uporabimo za shranjevanje točk(vozlišč) na platnu
* Sklad celih števil sklad – Del algoritma, vanj sproti dodajamo sosede trenutnega vozlišča
* Int start – začetno vozlišče(0)
* Seznam obiskani – Notri shranjujemo obiskana vozlišča.
* Seznam neobiskani\_sosedje – Neobiskani sosedje trenutnega vozlišča v algoritmu.
* Int counter – Preklopnik, ki omogoča spremljanje algoritma z uporabo 2 faz.
* Seznam helper – Seznam, ki hrani vozlišča, za vizualni prikaz.
* Int steviloVozlisc – število vozlišč, ki ga določimo v drugem Frame-u.