

Sortovi

- selection sort – traži najmanji element u nizu i zamjenjuje ga s prvim elementom
 - vrijeme izvođenja ne ovisi o početnom redoslijedu, nego o broju elem.
 - vrijeme izvođenja i u najboljem i najlošijem slučaju: $\theta(n^2)$
- bubble sort – zamijeni susjedne elemente ako nisu u dobrom redoslijedu
 - poboljšani: ako u nekom prolazu nije bilo zamjene, niz je sortirani (za poboljšani vrijedi $\Omega(n)$ i $O(n^2)$)
 - prosječan i najlošiji slučaj za obje varijante: $\theta(n^2)$
 - poboljšani za najbolji slučaj: $\theta(n)$ | za običan: $\theta(n^2)$
 - stabilno sortiranje
- insertion sort – dva dijela niza: sortirani i nesortirani
 - najbolji slučaj: sortirani niz $\theta(n)$ | najlošiji slučaj: naopako sortirani $\theta(n^2)$
 - vrijeme izvođenja: $\Omega(n)$ i $O(n^2)$
 - stabilno sortiranje
- Shellov sort – slično kao insertion sort, samo *skače* za zadani broj koraka po nizu
 - najlošiji slučaj $O(n^2)$
- Quicksort – koraci: 1. izračunamo stožerni element (ako se radi o medijanu 3 elemenata, odaberemo 1., srednji i posljednji element u nizu – srednji element je onaj „lijevi“ ako polje ima paran broj elem.)
 2. stožer *skrivamo* na predzadnje mjesto
 3. brojač **i** postavimo na prvi član, a brojač **j** na stožerni član
 4. **i** se miče udesno tako dugo dok ne nađe na element veći od stožera, analogno za **j** (prema lijevo dok ne nađe manji)
 5. elementi se zamjenjuju i brojači se nastavljaju šetati
 6. kad se **i** i **j** mimoiđu, stožer se vraća na mjesto **i**
 7. s lijeve strane imamo članove manje od stožera, a s desne veće
 8. ako je u zadatku zadan *cutoff* za određeni broj elemenata, primjenjujemo taj algoritam na lijevi/desni dio niza, a desni/lijevi dio niza opet sortiramo Quicksortom (idemo na korak 1.)
 - prosječno vrijeme izvođenja: $O(n \log n)$, najlošiji slučaj: $O(n^2)$

Stabla

- stupanj je broj podstabala nekog čvora -> stupanj stabla je najveći stupanj u stablu
- razina: korijen je razine 1, dijete korijena razine 2... -> dubina stabla je najveća razina
- binarno stablo – sastoji se od 0, 1 ili više čvorova stupnja 2
 - maksimalni broj čvorova na k -toj razini je 2^{k-1}
 - maksimalni broj čvorova dubine k je $2^k - 1$
 - stablo visine k i s $2^k - 1$ elemenata je puno stablo
 - stablo s n čvorova dubine k je potpuno ako čvorovi odgovaraju čvorovima punog stabla dubine k koji su numerirani od 1 do n
- obilazak stabla – inorder: lijevo -> korijen -> desno
 - preorder: korijen -> lijevo -> desno
 - postorder: lijevo -> desno -> korijen

Gomila

- prioritetni red je slična redu: skida se onaj podatak koji ima najveću vrijednost
- gomila je potpuno binarno stablo gdje se čvorovi mogu uspoređivati, npr. \leq
- gomilu s relacijom $>$ zovemo *max heap*, a s $<$ *min heap*
- heap sort – element s vrha gomile zamjenjuje se s posljednjim elementom polja
 - složenost podešavanja je $O(\log_2 n)$, a to se obavlja n puta pa je $O(n \log_2 n)$

Grafovi

- stupanj grafa ili valencija: broj bridova povezanih s vrhom (petlja se broji 2 puta)
- pravi brid ili karika: brid koji spaja različite vrhove
- orijentirani graf: usmjereni graf gdje između dva vrha postoji samo 1 usmjereni brid
- miješani graf: sastoji se od usmjerenih i neusmjerenih bridova
- multigraf: neusmjeren graf, dozvoljava višestruke bridove, ali ne i petlje
- pseudograf: multigraf koji dozvoljava i petlje
- put: šetnja koja ne uključuje niti jedan vrh 2 puta
- staza: šetnja koja ne prolazi dva puta istim bridom
- ciklus: put koji počinje i završava u istom vrhu
- krug: staza koja počinje i završava u istom vrhu
- ostale definicije i pojmovi identične u DISMAT
- obilasci: DFS i BFS: složenost obje $\Theta(|E| + |V|)$
- Dijkstrin algoritam

Raspršeno adresiranje

- kolizija: ako se više ključeva mapira na isto mjesto u hash tablici
- tehnike za rješavanje kolizije: ulančavanje, otvoreno adresiranje (linearno, kvadratno, dvostruko raspršeno)
- ulančavanje – elemente koji se preslikavaju na istu lokaciju dodajemo u listu
 - najgori slučaj: svaki ključ se mapira na isto mjesto
- otvoreno adresiranje – ne koristi se memorija izvan hash tablice (postoji mogućnost da se hash tablica popuni i da ne možemo dodavati nove elem.)
- linearno ispitivanje – $h(k, i) = (h'(k) + i) \bmod m$
 - problem linearnog grupiranja: ključevi se nakupljaju u pojedinim dijelovima tablice
- kvadratno ispitivanje – $h(k, i) = (h'(k) + c_1 i + c_2 i^2) \bmod m$
 - rješava se problem lin. grup., ali postoji problem kvadr. grup.
- dvostruko raspršeno – $h(k, i) = (h_1(k) + i h_2(k)) \bmod m$
 - ovisi o dvije funkcije pa se izbjegava problem grupiranja
 - vrijednost $h_2(k)$ mora biti relativno prost broj u odnosu na veličinu tablice m

Pretraživanje znakovnih nizova

- naivni: iscrpno pretražuje ulazni niz (kao „na papiru“)
- Rabin-Karp – računa sažetke podnizova, ako se podudaraju uspoređuje se podniz
- Knuth-Morris-Pratt – koristi se LPS polje