

Objašnjenje algoritama

- **Graf**

Da bi implementirali odgovarajući oblik ove strukture podataka koristili smo usmeren graf. Čvorove (vertex) smo implementirali tako da sadrže linkove HTML stranica, a njih smo povezali sa dve veze (edges) i to linkovima onih stranica na koje koje pokazuju – outgoing kao i linkovima stranica koje na njih pokazuju – incoming. Da bi popunili graf koristili smo linkove dobijene parsiranjem tako što smo svaki link upisivali u listu veza. Zatim smo pravili čvorove prolazeći kroz tu listu povezivajući ih na odgovarajući način u zavisnosti od toga ka kojim stranicama pokazuju, odnosno koje stranice ka njima pokazuju.

- **Set**

Da bi implementirali set koristili smo strukturu podataka dictionary koja se sastoji samo iz ključeva, bez vrednosti. Operacije koje su bile potrebne za set su unija, presek i komplement skupa. Uniju dva skupa smo implementirali tako što smo prošli kroz sve ključeve skupa A i njima smo dodali sve ključeve skupa B. Presek smo napravili tako što smo prolazili kroz ključeve skupova B i A, a one ključeve koji su se pojavljivali u oba skupa smo vratili kao rezultujući skup. Za rezultat komplementa skupa smo vratili one ključeve koji postoje u skupu A, a ne postoje u skupu B tako što smo prošli kroz ključeve skupa A i sve ih dodali, a zatim prošli kroz ključeve skupa B, i uporedivši ih sa onima u A obrisali iz rezultujućeg skupa sve one koji su se pojavili u skupu B.

- **Sort**

Kao algoritam za sortiranje korišćen je rekurzivni quick sort algoritam (podeli-pa-vladaj). On radi tako što izabere element koji predstavlja pivot i zatim podeli niz sa kojim radi oko tog pivot elementa. Postoji više verzija ovog algoritma ali mi smo koristili onu koja uvek bira krajnji element niza za pivot. Ključ ovog algoritma je podela niza, počinjemo od krajnjeg elementa i čuvamo podatak o indeksu manjeg (ili jednakog) elementa od pivota. Prolazimo kroz niz sa ciljem da sve elemente manje od pivota postavimo ispred njega, a ako naiđemo na element koji je veći preskačemo ga i nastavljamo dalje dok ne dođemo do kraja niza. Zatim dobijeni niz delimo na levi i desni deo, u zavisnosti od pivot elementa i ponovo sortiramo niz s tim što svaki element sa leve i desne strane postaju novi pivot elementi. Očekivano vreme izvršavanja ovog

algoritma je $O(n \log n)$, osim u najgorem slučaju kada je pivot najmanji odnosno najveći element i tada je $O(n^2)$.

- **Rangiranje**

Na rang stranice utiču tri komponente: broj ponavljanja traženih reči, broj linkova drugih stranica na tu stranicu i broj traženih reči u stranici koje sadrže link na tu stranicu. Mi smo osmislili kriterijum rangiranja po sledećoj formuli:

$$\text{RANG} = \text{BR_REČI_PO_STR} + \text{BR_LINKOVA_KA_STR} * 0,7 + \text{BR_REČI_NA_STR_SA_LINKOM} * 0,5$$