# Drugi projekat

Potrebno je napraviti mašinu za pretraživanje jednog dokumenta u pdf formatu (search engine). Program prilikom startovanja treba da obiđe pdf dokument, da parsira stranice u njemu i da izgradi strukture podataka potrebne za efikasno pretraživanje. Nakon toga, program omogućuje korisniku da unosi tekstualne upite koji se sastoje od jedne ili više reči razdvojenih razmakom, pretražuje stranice koristeći prethodno kreiranu strukturu podataka i korisniku ispisuje rangirane rezultate pretrage.

Kao test fajl koristiti knjigu *'Data Structures and Algorithms in Python'* (nalazi se u Files/Literatura). Postoji mogućnost testiranja i na test fajlovima koji će biti naknadno utvrđeni. Primer jednostavnog parsiranja možete pronaći u sekciji Files u folderu Projekat 2.

Preporučuje se parsiranje pdf fajlova upotrebom biblioteke **pdftotext**.

Rezultati treba da sadrže redni broj rezultata, redni broj stranice, kao i kratak kontekst (isečak iz dela stranice) u kome je tražena reč pronađena. Poželjno je da tražena reč (ili više njih) bude označena u isečku (npr. bojenjem konzolnog ispisa).

Ukoliko postoji problem sa učitavanjem pdf-a, mogu se koristiti i već parsirani txt fajlovi (iz Files/Projekat 2).

## Za maksimalnih 10 poena:

* Implementirati rangiranje rezultata pretrage tako da na rang rezultata utiče broj pojavljivanja traženih reči na stranici uz korišćenje proizvoljnih struktura podataka.
* Ukoliko korisnik unosi upit sastavljen od više reči, rangiranje stranica po svakoj pojedinačnoj reči utiče na sveukupno rangiranje određene stranice. U ovom slučaju, ne treba insistirati na prisustvu svake od reči u rezultatima, ali bi trebalo bolje rangirati rezultate u kojima se pojavljuju sve reči.

## Za više od 10 poena:

* Na rangiranje, osim broja pojavljivanja traženih reči na stranici, treba da utiče i broj veza sa drugih stranica kao i broj traženih reči na stranicama koji sadrže vezu na traženu stranicu. Veze ka stranicama se mogu prepoznati na osnovu napomena u tekstu (npr. 'See page 136', '... on page 87', see pages 34 and 35').
* Za organizovanje stranica koristiti graf. Za efikasnu pretragu reči na stranici koristiti strukturu podataka Trie.
* Potrebno je podržati ispravnu upotrebu logičkih operatora AND, OR i NOT prilikom formiranja upita uz kombinovanje operatora.  
  Primeri:  
   python AND sequence  
   dictionary NOT list  
   python OR dictionary NOT word
* Obezbediti paginaciju rezultata. Ograničiti broj ispisa po stranici na N (rezultati od 1 do N) uz nuđenje opcije za ispis sledećih N rezultata (od N+1 do 2N) itd.

## Za više od 25 poena:

* Upotreba fraza. Fraza se navodi pod navodnicima. U rezultatima se prikazuju (uz rangiranje) stranice u kojima se navedeni delovi fraze pojavljuju uzastopno u istom redosledu.
* Implementacija predlaganja alternativnih ključnih reči za pretragu *(did you mean)*. Ukoliko rezultata pretrage nema (nema rangiranih stranica) ili se zadati upit pojavljuje na malom broju stranica, ponuditi korisniku da zadati upit zameni sličnim, popularnijim upitom.

Druga unapređenja funkcionalnosti mogu doneti do 5 dodatnih poena. Za većinu dodatnih funkcionalnosti potrebno je istražiti mogućnosti drugih biblioteka za rad sa pdf dokumentima.

Primeri:

* 1 poen - Čuvanje rezultata pretrage u obliku pdf-a. Izdvojiti stranice prvih 10 rezultata i objediniti ih u poseban pdf fajl.
* 3 poena - Označiti (pdf highlighting) bojom pronađene ključne reči (kao u drugom pasusu) u pdf dokumentu sačinjenom objedinjavanjem stranica.
* Druge funkcionalnosti koje predmetni profesor ili asistent odobre.

## Dodatna objašnjenja:

Logički operator AND zahteva prisustvo svih navedenih reči u stranicama koje su u rešenju. Sve navedene reči bi trebalo u jednakoj meri da utiču na ukupan rang.

Logički operator OR zahteva prisustvo bar jedne od navedenih reči u stranicama koje predstavljaju rešenje.

Logički operator NOT se u ovom slučaju smatra BINARNIM dakle, predviđa se upotreba u obliku 'reč1 NOT reč2' gde se pojavljivanje reči *reč1* zahteva u rešenju dok se *reč2* u rešenju ne sme pojaviti.

**Opšte informacije:**

* Projektni zadatak nosi 30 poena.
* Fajl smestiti u folder nazvan po broju indeksa formata sw\_YY\_XXXX (npr. sw\_11\_2018)
* Zip arhivu gorenavedenog folder upload-ovati kao assignment na enastavu.
* Ukoliko bude problema sa upload-om, možete u predviđenom roku poslati zip na email adresu predmetnog asistenta.

Rok za slanje arhive je 23.06.2019.