

Uvod

December 18, 2024

1 Algoritmi in podatkovne strukture 1

Namen predmeta APS1 je naučiti udeležence *algoritmičnega razmišljanja*. Ukvarjali se bomo s *pravilnostjo* in *učinkovitostjo* algoritmov. Spoznali bomo več osnovnih algoritmov in z njimi povezanih podatkovnih struktur, ki bodo predstavljali našo osnovno orodjarno. Poleg tega bodo služili kot primeri, na katerih se bomo učili načrtovanja ter analiziranja algoritmov in podatkovnih struktur. S konkretnimi implementacijami zasnovanih idej pa bomo utrjevali in poglobljali znanje programiranja.

Algoritem - Postopek za reševanje določenega problema, ki ga reši v končnem številu korakov in je nedvoumno opisan s končnim številom ukazov, izvedljivih mehanično brez rabe uma in običajno podanih v obliki psevdokode, diagrama poteka ali v nekem programskem jeziku.

Primeri: iskanje največjega elementa v seznamu, Evklidov algoritem, Kruskalov algoritem, ...

Včasih med algoritme štejemo tudi nekoliko bolj dvoumne postopke opisane v naravnem jeziku, kot so npr. recepti, razni birokratski postopi (npr. postopek za prijavo začasnega prebivališča) itd.

Podatkovna struktura - Način organizacije podatkov, ki omogoča učinkovito izvajanje operacij na njih.

Primeri: seznam, množica, uravnoteženo iskalno drevo, ...

Algoritmi in podatkovne strukture se močno prepletajo. Algoritmi (postopki) v svojih korakih potrebujejo učinkovito organizirane podatke (podatkovne strukture). In obratno. Podatkovne strukture potrebujejo določene postopke (algoritme), da vzdržujejo podatke organizirane in omogočajo učinkovite operacije. Glede na delež enega ali drugega, govorimo o algoritmu ali podatkovni strukturi.

1.1 Izvajanje predmeta

Predmet se izvaja v tedenskih sklopih s predavanji, ki jim sledijo vaje. Na predavanjih so bomo ogledali teorijo in kakšen praktičen primer rešili tudi skupaj. Na vajah boste vsak teden samostojno (ob pomoči asistentov) reševali programersko nalogo v programskem jeziku C++, ki bo vključevala nek algoritmičen problem povezan z vsebino predavanj. Za domačo nalogo vas bo čakal podoben problem. Ob koncu sklopa se bodo vaše rešitve avtomatsko testirale. Za opravljeno nalogo mora vaša rešitev uspešno prestatiti vse primere. Prejeli boste povratno informacijo o številu uspešno prestalih testov, ne pa o njihovi vsebini. Kdor bo želel svojo rešitev popraviti, bo imel za to čas do konca naslednjega sklopa.

1.2 Delovno okolje

Vaše programe bomo ocenjevali na Ubuntu 22.04 s prevajalnikom GCC 11 in s standardom C++20. Če je vaš oddani program `resitev.cpp`, ga bomo prevedli in pognali z

```
g++ -std=c++20 -o program resitev.cpp
./program < test.in > test.res
```

Če si boste pripravljali virtualko z Ubuntu 22.04, morate dodatno namestiti `build-essential` paket, s katerim dobite C++ prevajalnik.

```
sudo apt install build-essential
```

1.3 Ocenjevanje

Za pristop k izpitu morate imeti uspešno opravljeno sprotno delo, kar pomeni povsem pravilno rešenih vsaj 50% tedenskih nalog. Skoraj pravilna rešitev je še vedno nepravilna in je zato ne upoštevamo. Oceno predmeta prejmete na izpitu, ki ga rešujete na papir.

Če odkrijemo dve ali več prepisanih rešitev, vsem udeleženiim ne priznamo sprotnega dela in tako v tekočem študijskem letu ne morete opravljati izpita. Zato ne objavljajte svojih rešitev. Posebej drzne kršitve bomo prijavili disciplinski komisiji.

1.4 Literatura

Algoritmi in podatkovne strukture:

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to algorithms. MIT press.
- Sedgewick R. & Wayne K. (2011). Algorithms fourth edition. Addison-Wesley.
- Aho A. V. Hopcroft J. E. & Ullman J. D. (1983). Data structures and algorithms. Addison-Wesley.
- Kononenko, I., Robnik Šikonja, M., & Bosnić, Z. (2008). Programiranje in algoritmi. Fakulteta za računalništvo in informatiko.

Programiranje v C++:

- [cplusplus](#), [cppreference](#)
- [Modern C++ for C Programmers](#)
- Stroustrup, B. (2013). The C++ Programming Language.-4th. Addison-Wesley.

1.5 Dodatne vaje

Na spletu je kup strani, ki omogočajo reševanje programerskih in algoritmičnih nalog (po tematikah) s preverjanjem pravilnost: [Codeforces](#), [SPOJ](#), [LeetCode](#), [HackerEarth](#), [HackerRank](#), ...