13. Algoritmusok bonyolultsága

Bonyolultság elmélet: Problémákat elemzünk, aszerint, hogy mennyire nehéz megoldani őket.

Probléma: Input (kérdés), Output (válasz), ha a kérdés eldöntendő, akkor döntési problémáról, ha nem keresési problémáról beszélünk. Megoldások ezekre a algoritmusok.

Inputméret.

* Gráfnál a gráfot megadó táblázat mérete, számnál a szám hossza.
* Jele: n

Lépésszám az inputméret függvényében

* f(n) az A algoritmus legnagyobb lépésszáma n hosszúságú input esetén

Polinomidejű algoritmus

* Egy algoritmus polinomiális, ha a lépésszáma felülről becsülhető a bemenet méretének polinomjával. (jó) – Pl.: összeadás, szorzás
* Egy algoritmus exponenciális, ha a futásideje exponenciális függvénye lehet a bemenet hosszának. (rossz) – Pl.: hatványozás

**Döntési problémák:**

* Adott bemenetre igen, vagy nem választ ad.
* Pl.: prímtesztelés: bemenete egy szám, kimenete egy bit, ami 1 ha prím, 0 ha nem.
* Pl.: Input:G(V,E) sEV output: összefüggő? Algoritmus: BFS

Bonyolultsági osztályok:

P

* Azok az A problémák, amikre található olyan algoritmus, mely minden inputra megoldja az A problémát, és mindezt polinom időben megoldja.
* legfeljebb (n) lépést végez
* pl.: Van-e Euler kör a gráfban, Gráf összefüggősége, síkbarajzolható-e a gráf

Np

co-NP

Feltételezett viszonyuk

Példa ilyen problémákra

Polinomiális visszavezethetőség (Karp-redukció)

NP-teljesség

Cook-Levin tétel

Nevezetes Np-teljes problémák

* SAT
* HAM
* 3-SZÍN
  + bizonyítás:
* k-SZÍN
  + bizonyítás:
* MAXFTN
  + bizonyítás:
* MAXKLIKK
  + bizonyítás: