

ChatGPT 4o

1. Algoritmusok bonyolultságának mérése, bonyolultságok típusai.  
Bonyolultsági függvények, ordó jelölés. Keresési és rendezési algoritmusok, gráfalgoritmusok bonyolultsága.

Algoritmusok bonyolultságának mérése

- Időbonyolultság: Az algoritmus futási idejének mértéke a bemeneti adatok méretének függvényében.
  - Általában az algoritmus által végrehajtott alapvető műveletek számával mérjük.
  - Különböző esetekre szokás mérni:
    - Legjobb eset: A lehető legkedvezőbb futási idő.
    - Legrosszabb eset: A lehető legrosszabb futási idő.
    - Átlagos eset: Az átlagos futási idő a bemenetek valamilyen eloszlása esetén.
- Tárbonyolultság: Az algoritmus által felhasznált memória mértéke a bemeneti adatok méretének függvényében.

Bonyolultságok típusai

- Konstans időbonyolultság ( $O(1)$ ): Az algoritmus futási ideje független a bemenet méretétől.
- Logaritmikus időbonyolultság ( $O(\log n)$ ): Az algoritmus futási ideje logaritmikusan növekszik a bemenet méretével.
- Lineáris időbonyolultság ( $O(n)$ ): Az algoritmus futási ideje arányosan nő a bemenet méretével.
- Polinomiális időbonyolultság ( $O(n^k)$ ): Az algoritmus futási ideje a bemenet méretének valamilyen hatványával nő.
- Exponenciális időbonyolultság ( $O(2^n)$ ): Az algoritmus futási ideje exponenciálisan nő a bemenet méretével.
- Faktoriális időbonyolultság ( $O(n!)$ ): Az algoritmus futási ideje faktoriálisan nő a bemenet méretével.

Üzenet a következőnek: ChatGPT

ChatGPT can make mistakes. Check important info.

ChatGPT 4o

- Tárbonyolultság: Az algoritmus által felhasznált memória mértéke a bemeneti adatok méretének függvényében.

Bonyolultságok típusai

- Konstans időbonyolultság ( $O(1)$ ): Az algoritmus futási ideje független a bemenet méretétől.
- Logaritmikus időbonyolultság ( $O(\log n)$ ): Az algoritmus futási ideje logaritmikusan növekszik a bemenet méretével.
- Lineáris időbonyolultság ( $O(n)$ ): Az algoritmus futási ideje arányosan nő a bemenet méretével.
- Polinomiális időbonyolultság ( $O(n^k)$ ): Az algoritmus futási ideje a bemenet méretének valamilyen hatványával nő.
- Exponenciális időbonyolultság ( $O(2^n)$ ): Az algoritmus futási ideje exponenciálisan nő a bemenet méretével.
- Faktoriális időbonyolultság ( $O(n!)$ ): Az algoritmus futási ideje faktoriálisan nő a bemenet méretével.

Bonyolultsági függvények, ordó jelölés

- Bonyolultsági függvény: Az algoritmus futási idejét vagy tárigényét leíró matematikai függvény.
- Ordó jelölés (Big-O notation): Az algoritmus futási idejének vagy tárigényének felső határát jelöli a bemenet méretének függvényében.
  - Példák:
    - $O(1)$ : Konstans időbonyolultság.
    - $O(n)$ : Lineáris időbonyolultság.
    - $O(n \log n)$ : Lineáris-logaritmikus időbonyolultság.
    - $O(n^2)$ : Négyzetes időbonyolultság.
    - $O(2^n)$ : Exponenciális időbonyolultság.

Keresési algoritmusok bonyolultsága

Üzenet a következőnek: ChatGPT

ChatGPT can make mistakes. Check important info.

