

C2142: Otázky k opakování

1. Je algoritmus se složitostí $\mathcal{O}(n)$ vždy lepší než algoritmus se složitostí $\mathcal{O}(n \log n)$? Zdůvodněte.
2. Ukažte, že selection sort není stabilní řadící algoritmus.
3. Jak funguje in-order DFS na binárním stromě? Co je jeho výsledkem na BVS?
4. Popište hlavní myšlenku Strassenova algoritmu. Do jaké kategorie metod spadá?
5. Jak a s jakou složitostí naleznete v minimové haldě druhý nejmenší prvek?
6. Popište myšlenku hladových algoritmů. Uveďte alespoň dva příklady takovýchto algoritmů.
7. Popište na konkrétním příkladě výhody obousměrně zřetěženého seznamu oproti jednosměrnému. Jaké jsou nevýhody?
8. Mějme dvě seřazené posloupnosti, každá délky $n/2$. Popište, jak pracuje algoritmus, který je sloučí do jedné seřazené. Jaká je jeho složitost? Kde se využívá?
9. Nakreslete jeden příklad rotace v AVL stromě. K čemu se rotací využívá?
10. Popište algoritmus, který určí vstupní stupeň vrcholu v orientovaném grafu reprezentovaném maticí sousednosti. Jaká je jeho složitost?
11. Na jakém principu funguje Primův algoritmus? Demonstrujte na netriviálním příkladu.
12. Ukažte, že rozhodovací varianta problému obchodního cestujícího je v třídě NP.
13. Popište algoritmus, který určí, zda jsou dvě slova přesmyčky (obsahují stejný počet stejných písmen). Tedy např. *LUSK* a *KLUS* jsou přesmyčky, *HLAD* a *SKLAD* ne. Jaká je jeho složitost?
14. Napište algoritmus na bázi backtrackingu, který určí, zdali je možné vyplatit zadanou částku mincemi daných hodnot. Vstupem je částka C a hodnoty mincí c_1, \dots, c_k . Počet mincí každé hodnoty je neomezený.
15. Jak a s jakou složitostí naleznete ve zleva zarovnané minimové haldě reprezentované v poli *největší* prvek?
16. Jakým algoritmem byste vypsali binární strom po jednotlivých úrovních? Jaká je jeho složitost?
17. Co je to kolize při hashování a jakým způsobem se dá řešit?
18. Popište myšlenku metody branch & bound. Uveďte alespoň jeden konkrétní příklad, kde by se jí dalo využít.
19. Diskutujte rozdíly mezi polem a spojovým seznamem.
20. Popište merge sort a odvoďte jeho složitost. (Pomožte si obrázkem.)
21. Na jakém principu funguje Kruskalův algoritmus? Demonstrujte na netriviálním příkladu.
22. Popište algoritmus, který určí výstupní stupeň vrcholu v orientovaném grafu reprezentovaném seznamem následníků. Jaká je jeho složitost?
23. Popište Bellman-Fordův algoritmus, ukažte jak probíhá výpočet na libovolném grafu s pěti vrcholy a sedmi hranami.
24. Ukažte, že rozhodovací varianta problému batohu je ve třídě NP. (Lze vložit do batohu nosnosti M předměty s celkovou hodnotou alespoň V ? Předměty mají hmotnosti m_1, \dots, m_k a hodnoty v_1, \dots, v_k .)
25. Co je to kostra grafu? Jakými algoritmy se dá spočítat minimální kostra grafu?

26. Jak zjistíte nejkratší vzdálenost v neohodnoceném grafu? Jaká je složitost vámi vybrané metody?
27. Popište, jak vypadá ideální a nejhorší hashovací funkce.
28. Ukažte příklad problému a jeho konkrétní instance, u které hladový algoritmus nedá (optimální) řešení.
29. Popište dva způsoby, jak lze reprezentovat zásobník. Jaká je v těchto případech složitost jednotlivých operací?
30. Uveďte rekurzivní a nerekurzivní podobu programu, který spočítá faktoriál přirozeného čísla. Je zde rozdíl ve složitosti obou přístupů?
31. Popište algoritmus Selection sort a odvoďte složitost.
32. Popište algoritmus, který do orientovaného neohodnoceného grafu reprezentovaného maticí sousednosti přidá opačné hrany, tj. pokud je v grafu hrana (u, v) a neexistuje hrana (v, u) , tak ji algoritmus doplní.
33. Uveďte příklad problému, který lze řešit jak hrubou silou, tak přístupem rozděl a panuj. Nastíňte hlavní myšlenku obou algoritmů.
34. Popište algoritmus, kterým byste určili počet listů binárního stromu. Jaká bude jeho složitost?
35. Vysvětlíte, v čem spočívá problém P vs. NP.
36. Popište algoritmus na bázi backtrackingu, který určí, zda má zadaná konfigurace Sudoku řešení.
37. Co je to graf a jak jej lze reprezentovat? Ukažte na příkladu.
38. K čemu je algoritmus binárního půlení? Jaká je jeho složitost? Jaké jsou vstupní podmínky pro tento algoritmus.
39. Co znamená, pokud je řadící algoritmus stabilní? Jak se dá tato vlastnost využít? Uveďte příklad stabilního řadícího algoritmu.
40. Jaký je rozdíl mezi prohledáváním do šířky a do hloubky? Ukažte na příkladu.
41. Existuje v grafu vždy pouze jedna minimální kostra? Pokud ne, uveďte příklad takového grafu.
42. Co je to prioritní fronta a jak byste ji implementovali? Jaká by byla složitost operací nad takovou strukturou?
43. Máte dán algoritmus pro výpočet nejkratší vzdálenosti v ohodnoceném *úplném* grafu. Můžete tento algoritmus použít na obecném grafu? Jak?
44. Zhodnote výhody a nevýhody rekurzivních algoritmů.
45. Popište Dijkstrův algoritmus. Ukažte jak probíhá výpočet na libovolném grafu s pěti vrcholy a sedmi hranami.
46. Popište obecný binární vyhledávací strom. Jaké standardní operace jsou na něm definovány? Čím je dána složitost těchto operací? Uveďte konstrukci nejnejpříznivějšího případu.
47. S využitím Master theoremu odvoďte složitost následujících algoritmů:
 - a) merge sort
 - b) vyhledávání binárním půlením
48. Proč nelze aplikovat Master theorem na quick sort?
49. Odvoďte, jaká je výška zleva zarovnané binární haldy o n prvcích.
50. Diskutujte rozdíly ve složitosti přidávání prvku do dynamického pole v závislosti na zvolené pozici.

51. Kdy se hodí využít při analýze algoritmu amortizovanou složitost?
52. Jak lze reprezentovat binární haldu v poli? Je vaše zvolená reprezentace vhodná pro libovolnou binární haldu?
53. Čím je (z velké části) dána složitost operací nad stromy? Uvedte příklad i s odvozením složitosti.
54. Popište datovou strukturu trie. Jak se liší od jiných struktur (např. binárního vyhledávacího stromu)?
55. Co si představíte pod pojmem *data structure augmentation*?
56. Navrhnete datovou strukturu pro reprezentaci disjunktních podmnožin. Jaké operace byste nad touto strukturou očekávali?
57. Vyjádřete počet kroků nutných pro vyřešení problému Hanoiských věží.
58. Jaký řadící algoritmus byste použili, pokud by záměna dvou prvků byla velmi drahá operace. Zdůvodněte.
59. Srovnejte přístup rozděl a panuj s dynamickým programováním.
60. Vysvětlete, kdy je vhodné užít heuristik. Uvedte příklad.
61. Uvažme dvě varianty algoritmu quick sort. První vybírá jako pivota první prvek posloupnosti, druhá poslední. Má takto rozdílná volba pivota vliv na korektnost nebo na složitost tohoto algoritmu?
62. Upravme merge sort tak, že při dosažení určité velikosti podposloupnosti použije pro její seřazení jednoduchý stabilní řadící algoritmus, např. insertion sort. Jaká je složitost tohoto řešení, pokud se insertion sort použije pro posloupnosti velikosti:
 - a) 16
 - b) $n/16$