(M)RAM

Úloha 1 *Napíšte RAM pre problém triedenia. Začnite s jednoduchým (z hľadiska programovania, nie efektívnosti) algoritmom Bubblesort. Určte zložitosť Vášho programu pri jednotkovej aj logaritmickej cene.*

Úloha 2 *Napíšte RAM pre problém Insertsort. Určte zložitosť Vášho programu pri jednotkovej aj logaritmickej cene.*

Úloha 3 *Napíšte (M)RAM, ktorý načíta "reťazec"—postupnosť čísel:* $n, a_1, \ldots, a_n$*, kde* $a_i \in \{1,2\}^*$*. Na výstup vypíše:*

1. $a_2, a_1, a_4, a_3, \ldots, a_n, a_{n-1}$ *alebo* $a_2, a_1, a_4, a_3, \ldots, a_{n-2}, a_{n-1}, a_n$ *podľa toho, či je* n *párne alebo nepárne*
2. *1 ak tvorí vstupná postupnosť palindróm* ($a_1 \ldots a_n = a_n \ldots a_1$)
3. *1 ak vstupná postupnosť obsahuje ako súvislý podreťazec palindróm dĺžky aspoň* $n/2$
4. *1 ak číslo, ktorého desiatkovým zápisom je vstupná postupnosť, je prvočíslo*

Určte zložitosť každého programu pri jednotkovej aj logaritmickej cene.

Turingove stroje

Úloha 4 *Napíšte TS, ktorý rozhoduje jazyk*

- $L = \{wcw \mid w \in \{a,b\}^*\}$
- $L = \{ww \in \{a,b\}^*\}$

Úloha 5 *Napíšte Turingove stroje pre problémy z úlohy 3*

Dynamické programovanie

Úloha 6 *Napíšte program, ktorý pre vstupnú postupnosť rozmerov* n *matíc vypočíta, v akom poradí treba matice násobiť, aby pre daný algoritmus násobenia dvoch matíc bola výsledná zložitosť optimálna. Aká je časová a pamäťová zložitosť Vášho algoritmu?*

Úloha 7 *Naprogramujte algoritmus konštrukcie OBVS z prednášky a analyzujte jeho časovú a priestorovú zložitosť.*

Modifikujte algoritmus za predpokladu, že navyše poznáme aj početnosť vyhľadávania prvkov, ktoré sa v množine, reprezentovanej BVS, nenachádzajú. Teda keď poznáme $q(i)$*– početnosť vyhľadávania prvku, ktorý je medzi* a_{i-1} *a* a_i*. Pritom predpokladáme, že* $q(1)$ *je početnosť vyhľadávania prvkov menších ako* a_1*,* $q(n+1)$ *je početnosť vyhľadávania prvkov väčších ako* a_n*.*

Úloha 8 *Nech* C *je matica cien ohodnoteného grafu s vrcholmi* $1, 2, \ldots, n$*;* $C[i, j]$ *udáva kladnú dĺžku hrany medzi vrcholmi* i *a* j*. Použite metódu dynamického programovania na výpočet matice* S *dĺžok najkratších ciest;* $S[i, j]$ *bude cena najkratšej cesty z vrchola* i *do vrchola* j*, pričom pod dĺžkou cesty rozumieme súčet cien hrán, ktoré cestu tvoria.*

Úloha 9 *Uvažujme problém hľadania* najdlhšej spoločnej podpostupnosti *dvoch postupností* $X = X_1, \ldots, X_n$, $Y = Y_1, \ldots, Y_m$*, ktorá vznikne vynechaním niektorých znakov z každej z nich. Označme* $LCS(i, j)$ *dĺžku najdlhšej spoločnej podpostupnosti postupností* $X_1, \ldots, X_i$ *a* $Y_1, \ldots, Y_j$*; zujíma nás zrejme* $LCS(n, m)$*. (LCS(ABAKUS, AUTOBUS)=4; najdlhšia spoločná podpostupnosť je ABUS)Základom algoritmu pre výpočet* $LCS(n, m)$ *založenom na dynamickom programovaní je nasledujúci vzťah:*

$$LCS(i, j) = \begin{cases} 1 + LCS(i-1, j-1), & \textit{ak } X_i = Y_j \\ \\ \min\{LCS(i-1, j), LCS(i, j-1)\}, & \textit{inak} \end{cases}$$

Napíšte program, ktorý nielen vypočíta $LCS(n, m)$*, ale aj príslušnú najdlhšiu spoločnú podpostupnosť. Aká je zložitosť Vášho programu?*

Úloha 10 *Majme ohodnotený graf* $G = (V, E)$*, zadaný maticou cien:* $C[i, j]$ *je cena hrany z vrchola* i *do vrchola* j*. Našou úlohou je nájsť takú permutáciu* π *vrcholov* $2, 3, \ldots, n-1$*, ktorá minimalizuje dĺžku kružnice* $1, \pi, 1$*; pod cenou kružnice rozumieme súčet cien jej hrán. Problému sa hovorí problém obchodného cestujúceho (TSP)*

Nech $S = \{1, 2, \ldots, n\}$, $j \in S$*. Označme* $C(S, j)$ *dĺžku najkratšej cesty* $1\pi j$*, kde* π *je permutácia množiny* $S - \{1, j\}$*.*
Základom riešenie problému TSP dynamickým proramovaním je nasledujúci vzťah

$$C(S, j) = \min_i \{C(S - \{j\}, i) + C[i, j]\}$$

Napíšte program, ktorý rieši TSP dynamickým programovaním. Aká je zložitosť Vášho programu?

Greedy

Úloha 11 *Za akých podmienok dáva greedy metóda optimálne riešenie pre problém mincí? Dokážte.*