

Řešení úlohy č. 2

Klece

Algoritmus. Náš algoritmus si nejprve seřadí zvířata podle jejich dravosti. Zvíře s nejmenší dravostí umístíme do prvního pavilonu a zbývajících zvířata budeme postupně zpracovávat ve vzestupném pořadí podle jejich dravosti. Při zpracovávání i . zvířete se podíváme, jestli jej můžeme vložit do pavilonu s předchozím zvířetem. To zkontrolujeme tak, že rozdíl dravosti nejkrotšího zvířete v pavilonu a současného zvířete nepřekračuje p . Pokud tato kontrola uspěje, pak zpracovávané zvíře přidáme do uvažovaného pavilonu. Jinak založíme nový pavilon a vložíme do něj právě zpracovávané zvíře. Takto pokračujeme, dokud nezpracujeme všechna zvířata.

Správnost. Nechtě P_1, \dots, P_k jsou pavilony, které náš algoritmus vytvoří. Pro $i \in \{1, \dots, k\}$ označme $x_i = \min P_i$. Zvířata s dravostmi x_1, \dots, x_k musí žít v různých pavilonech, neboť rozdíl libovolných dvou těchto čísel je větší než p . Z toho však dostáváme, že každé řešení musí vypsat alespoň k různých klecí, přičemž náš algoritmus jich vypíše přesně k . To znamená, že vypsané řešení je optimální.

Časová složitost. Algoritmus si nejprve seřadí dravosti všech zvířat. To se zvládne v čase $O(n \log n)$ – například algoritmem `merge_sort`. Dále algoritmus zpracovává jedno zvíře po druhém. Pro ušetření času si bude pamatovat dravost nejkrotšího zvířete, které žije v pavilonu zvířete, které předchází zpracovávanému zvířeti. Tím pádem zpracování všech zvířat zvládneme v lineárním čase. Dohromady dostáváme časovou složitost $O(n \log n)$.