Řešení 4. úlohy 2. kola - Klece

Úvod do problému

Naším úkolem je roztřídit zvířata do klecí tak, aby se nepožrala, požerou se tehdy, když jejich rozdíl jejich agresivit bude větší než nějaký limit `p`. Matematicky řečeno: musíme rozdělit množinu čísel do minimálního počtu podmnožin tak, aby rozdíl minima a maxima podmnožiny by menší nebo roven `p`.

Obecný popis řešení

Základem mého řešení je setřídit zvířata vzestupně podle agresivity. Následně budeme procházet seznam zvířat s tím, že si zapamatujeme první zvíře a hledáme nejvzdálenější zvíře které to první nesežere. Když ho najdem tak dáme všechna zvířata, mezi prvním a tím, kde právě jsme, do jedné klece. Další zvíře si zase zamapatujeme a zase hledáme nejvzdálenější zvíře, které to zapamatované nesežere. Když ho najdem, tak zase dáme všechny zvířata mezi zapamatovaným a nalezeným do jedné klece. Takhle pořád do kola, dokud nedojdeme na konec seznamu zvířat. Potom budeme mít všechna zvířata v klecích.

Implementace

Třízení

Na třízení máme několik algoritmů, mezi nejčastěji užívané patří quicksort a mergesort. Jejich výhoda je, že jsou univerzální a relativně rychlé a dají se použít na jakýkoli vstup. Quicksort má průměrně `O(n*log(n))` časovou složitost, ale `O(n^2)` nejhorší a `O(log(n))` paměťovou složitost. Merge sort bude mít časovou složitost `O(n*log(n))` ve všech případech, ale jeho paměťová složitost `O(n)` by mohla být lepší. Ale protože my vstup známe, tak nejsme limitováni na tyto univerzální algoritmy a můžeme použít třeba radix sort jehož limitací je, že potřebujeme znát počet cifer čísel, které budeme třídit, to pro nás ale není problém, to si můžeme zapamatovat během čtení vstupu. Stejně to musíme udělat, tak proč toho nevyužít. Radix sort má časovou složitost i paměťovou složitost `O(n)`. Funguje na následujícím principu: seřadí všechna čísla nahází do kyblíků podle poslední cifry, z těchto kyblíků je následně vytahuje (v pořádí v jakém se do kbelíku dostala) od nejmenší cifry a vkládá je zpět do seznamu. Tohle opakuje pro všechny cifry. Na konci pak má seřazený seznam.

Rozdělování do klecí

Rozdělení do klecí provedeme následovně. Vytvoříme si nový seznam klecí a postupně budeme procházet přes všechna zvířata, která si budeme přidávat do pomocného seznamu. Když při procházení narazíme na zvíře, které by sežralo to první na pomocném seznamu, tak pomocný seznam uložíme jako klec do seznamu klecí a vytvoříme si nový pomocný seznam. A pokračujeme dál v iteraci dokude nedojdeme na konec.

Důkaz o správnosti algoritmu

Algoritmus najde jedno z možných řešení. Toto řešení bude správné, protože: Zvířata jsou seřazena podle velikosti tudíž, kdybychom chtěli 1. klec zvětšit, tak musíme na začátek přidat další klec, protože by jinak zvíře, které jsme přidali sežralo to nejmíň agresivní. Kdybychom chtěli 1. klec zmenšit, tak bychom museli zvětšit tu další, ale to nejde, protože bychom jí museli na druhém konci zmenšit. V některých případech se stane, že i

když ji posuneme, tak se nám pořád do poslední klece vejdou všechna zvířata, protože správných řešení je několik, ale v některých dojde k tomu, že bychom museli přidat další klec.

Složitost

Časová složitost algoritmu závisí na zvoleném třídícím algoritmu, v tomto případě je `O(n)`, k tomu musíme přičíst časovou složitost samotného rozřazování do klecí, které sestává z jednoho průchodu všemi zvířaty, proto je také `O(n)`, výsledné `O(2n)` se pokrátí na `O(n)`. S paměťovou složitostí je to to samé. Třídící algoritmus `O(n)` + rozdělování `O(n)` = `O(n)`.