Übungen zur Algorithmischen Bioinformatik I

Blatt 5

Xiheng He

Mai 2021

1. Aufgabe: Geschlossene Probleme (closed problems) (10 Punkte)

(a) In der Vorlesung haben Sie verschiedene Algorithmen zum Sortieren von Listen von Elementen bei gegebener Vergleichsoperation ($<, \leq$) auf den Elementen kennengelernt (= Problem SORT).

Ist das SORT-Problem closed?

SORT-Problem ist closed.

Erklären Sie was closed bedeutet!

Wenn die entsprechende Algorithmen für solche Probleme einerseits eine inhärente Laufzeit-Komplexität (z.B. Mergesorts is $O(n \log n)$) und andererseits eine untere Schranke $\Omega(n \log n)$ haben, sind diese Probleme "closed problem".

Beweisen oder widerlegen sie closed(SORT)!

Gegeben sei genau n unterschiedliche Zahlen. In best-case wurden sie alle schon sortiert damit wird dies nicht betrachtet. In worst-case sind alle nicht sortiert dann nehmen wir an, dass die Algorithmen den richtige Ergebnis liefern nur dann wenn sie x Schritte berechnet haben nämlich 2^x Vergleichsoperationen durchführen müssen, weil für jeden Schritt zwei Elemente verglichen werden deshalb es nur zwei mögliche Fälle für jeden Vergleich gibt.

Ferner sind theoritsch auch $n!$ Permutationen für solche Sotierprobleme vorhanden und Algorithmen benötigen höchstens 2^x Vergleichsoperationen um die einzige richtige Permutation herauszufinden. Deshalb folgt:

$$
\begin{aligned}
&2^x \geq n! \implies x \geq \log(n!) \\
&\because n! = n(n-1)(n-2)\dots\frac{n}{2}\dots 1 \geq (\frac{n}{2})^{\frac{n}{2}} \\
&\therefore \log(n!) \geq \log((\frac{n}{2})^{\frac{n}{2}}) \\
&= \frac{n}{2} \cdot \frac{\log n}{\log 2} \\
&= \Theta(n \log n) \\
&\implies x = \Omega(n \log n)
\end{aligned}
$$

Daraus können wir Schlussfolgerungen ziehen, dass die untere Schranke für Sortieralgorithmen $\Omega(n \log n)$ ist.

Zusammengefasst ist eine obere Schranke $O(n \log n)$ für SORT-Problem schon bekannt und eine untere Schranke $\Omega(n \log n)$ schon beweist dann wird solche Problem als "closed-problem" genannt.

(b) Auch das Traveling Salesperson Problem (TSP) kennen Sie z.B. Aus der Vorlesung.

Ist closed(TSP)?

TSP ist nicht closed.

Beweisen oder widerlegen sie closed(TSP)!

TSP ist als “NP-hard” Problem bekannt, d.h. kein Algorithmus existiert, der eine optimaleste Reiseroute in polynomieller Laufzeit bestimmt. Es gibt zwar konkrete Algorithmen, z.B berechnen alle Permutationen mit $O(n!)$ Komplexität und Held–Karp algorithm mit $O(n^2 2^n)$ Komplexität aber wir können kein k festlegen, ob ein Algorithmus mit $\Omega(n^k)$ vorhanden ist. Daher existiert lower-bound nicht. Weiterhin können wir ein theoritsches lower-bound $\Omega(n^2)$ herleiten, da der TSP Algorithmus mit $n(n-1)/2$ Entfernungen für n Städten umgehen muss jedoch ist dies sinnlos da wir noch keinen Algorithmus in polynomieller Laufzeit gefunden haben.