Karlsruher Institut für Technologie
Fakultät für Informatik
ITI Wagner

Algorithmische Geometrie
Sommersemester 2011
Martin Nöllenburg/Andreas Gemsa

Übungsblatt 4 - Lineare Programmierung

Ausgabe: 03. Mai 2011
Abgabe: 10. Mai 2011

1 Korrektheitsbeweis

In der Vorlesung wurde der Algorithmus RandomPermutation(A) vorgestellt.

a) Beweise seine Korrektheit, indem du zeigst, dass jede mögliche Permutation von A gleich wahrscheinlich ist.

b) Zeige außerdem, dass die Aussage aus a) nicht mehr stimmt, wenn wir $r \leftarrow \text{Random}(k)$ durch $r \leftarrow \text{Random}(n)$ ersetzen.

2 Paranoia

Algorithmus 1: ParanoidMax

```
Eingabe : Endliche Menge A ⊂ ℝ
Ausgabe : Maximum max_{a∈A} a der Menge
if |A| = 1 then
    return einziges Element a ∈ A
else
    a = zufällig gewähltes Element aus A
    b = ParanoidMax(A \ {a})
    if b ≥ a then
        return b
    else
        prüfe unnötigerweise jedes Element aus A \ {a}, um sicherzugehen, dass a wirklich
        größer ist
        return a
```

Betrachte Algorithmus 1, der das Maximum einer Menge von Zahlen berechnet. Schätze die asymptotische Worst-Case-Laufzeit des Algorithmus scharf ab. Beachte die zufällige Auswahl des Elementes a und zeige, dass die erwartete Laufzeit echt besser ist.

Bitte umblättern

3 Züge

Gegeben seien n Züge, die auf parallelen Gleisen fahren. Jeder Zug z_i ($i = 1, \dots, n$) fahre dabei mit konstanter Geschwindigkeit v_i und nehme zum Startzeitpunkt $t = 0$ (hier startet nur die Zeit, die Züge haben schon die volle Geschwindigkeit) die Position k_i auf der Strecke ein. Gebe einen Algorithmus an, der in $O(n \log n)$ Zeit berechnet, welche Züge bis zu einem gegebenen Zeitpunkt $t_{\text{stop}} > 0$ mindestens einmal in Führung waren.

Hinweis: Der in der Vorlesung vorgestellte Algorithmus zu Berechnung vom Schnitt mehrerer Halbebenen könnte dabei nützlich sein.