問題（k 番目に小さな要素を見つける問題）

各要素が異なる入力配列（長さ n）における k 番目に小さな要素を見つけるアルゴリズム SELECT の計算性能を考える。

SELECT

A) 入力配列を 5 つの要素からなるグループへ分割。残った要素がある場合、1 つのグループにする。グループ数は $\lceil n/5 \rceil$ 個。ただし $\lceil x \rceil$ は実数 x 以上の最小の整数で、たとえば$\lceil 27/5 \rceil = 6$。

B) 各グループの中央値（median）を計算。

C) $\lceil n/5 \rceil$ 個の中央値全体の中央値 x （median of medians）を、SELECT を再帰的に実行して計算。ただし、$\lceil n/5 \rceil$ が奇数のとき中央値は 1 つに定まるが、偶数の場合は中央に 2 つの候補があるため、昇順で $\lceil n/5 \rceil/2$ 番目の値を中央値と定義する。奇数、偶数どちらの場合も、昇順で$\lceil \lceil n/5 \rceil/2 \rceil$番目の値が中央値となる。

D) x 未満と x 以上のブロックへ分割し、k 番目の要素があるブロックを決定。

E) k 番目の要素があるブロックを SELECT でさらに検索。

以下の設問に答えよ。

(1) x より大きい要素は $3n/10 - 6$ 個以上あることを証明せよ（同様に x より小さい要素は $3n/10 - 6$ 個以上ある）。よってステップ E で SELECT が検索するブロックの長さは高々 $7n/10 + 6$ である。

(2) 長さ n の入力配列を SELECT が処理する最悪時間計算を $T(n)$ と記述するとき、$T(n)$ が満たす条件式を示せ。ただしステップ A,B,D は線形時間で計算できると仮定してよい。

(3) n が十分に大きい時、$T(n)$ が $O(n)$ の関数であること、すなわち SELECT が線形時間アルゴリズムであることを示せ。

証明は

Cormen et al. “Introduction to Algorithms, third edition” (MIT Press)

9. Medians and Order Statistics. 9.3 Selection in worst-case linear time

を日本語で解説した内容。

(1) x より大きい要素は $3n/10-6$ 個以上あることを証明せよ。

$\lceil n/5 \rceil$ 個の中央値全体の中央値 x （median of medians）は、昇順で$\lceil \lceil n/5 \rceil /2 \rceil$番目のブロック A の中央値であることは問題文中に書いています。

x より大きい要素は、このブロック A の次のブロックから最後から２番目のブロックまでに各々３個は存在する。なお、ブロック A は x より大きい要素を２個しか含まないので除外する。最後のブロックも、要素数が３以下の場合 x より大きい要素を３個未満しか含まない可能性もあるので除外する。少し過剰に除外する場合もあるが、x より大きい要素の総数は、少なくとも以下の不等式をみたす。

$$3(\lceil \lceil n/5 \rceil /2 \rceil -2) \geq 3n/10-6$$

(2) 長さ n の入力配列を SELECT が処理する最悪時間計算を $T(n)$ と記述するとき、$T(n)$ が満たす条件式を示せ。

n が十分に大きい場合（最悪計算量の見積の際には、このような大雑把な言い方で議論を進めるが、何故そんなことをして良いのかは後で説明)、以下の式が成り立つ。

$$T(n) \leq T(\lceil n/5 \rceil) + T(7n/10+6) + an$$

$T(\lceil n/5 \rceil)$ はステップ C で $\lceil n/5 \rceil$ 個の中央値全体の中央値 x （median of medians）を、SELECT を再帰的に実行して計算する際の計算時間。$T(7n/10+6)$ はステップ E で SELECT が検索する計算時間（ブロックの長さは高々 $7n/10+6$ なので)。an はステップ A,B,D の計算時間の総計の上界（an が上界になるように a は十分に大きな定数を選ぶとする)。

(3) n が十分に大きい時、$T(n)$ が $O(n)$ の関数であること、すなわち SELECT が線形時間アルゴリズムであることを示せ。

$T(n) \leq cn$ が満たすことを n の大きさに関する帰納法で示す。cn が上界になるように、これから定数 c の値は決めてゆく。 帰納法なので本来であれば、n がある定数 n_0 以下の場合を証明した後で、 n が十分に大きい $n > n_0$ の場合の証明に進む。しかし、このあ

る定数 n_0 をどのぐらいに設定すればよいかを調べたいので、その見通しを立てるために、まず不等式が成立つような n が十分に大きい状況を考える。

$T(n)$

$\leq T(\lceil n/5 \rceil) + T(7n/10 + 6) + an$

$\leq c(n/5 + 1) + c(7n/10 + 6) + an$　帰納法の仮定と$\lceil n/5 \rceil \leq n/5 + 1$

$= 9cn/10 + 7c + an = cn + (-cn/10 + 7c + an)$

$T(n) \leq cn$ になってほしいが、そのためには

$(-cn/10 + 7c + an) \leq 0$

でないと困る。上の条件が成り立つためには n を十分に大きくすればよいものの、どのぐらい大きくする必要があるのか？　定数 c と a の大きさも調整する必要があるので、上の不等式と等価な以下の不等式へと変形する（ただし条件 $n \geq 71$ のもと）。

$c \geq 10a(n/(n - 70))$

この不等式を成り立たせるために、条件 $n \geq 71$ のもと $c \geq 710a$ となるように定数 c の値をとる（あとから定数の大きさを考える辻褄をあわせる論法)。この $n \geq 71$ こそが「 n が十分に大きい場合」であり、帰納法の仮定 $T(n) \leq cn$ が

$$T(n) \leq T(\lceil n/5 \rceil) + T(7n/10 + 6) + an$$

を満たす場合である。一方逆に、 n が「小さい」場合、つまり $n \leq 70$ のときに $T(n) \leq cn$ が成り立つかどうかについて考える。これまでの議論では $c \geq 710a$ となるように定数 c を取れるのであれば、c はいくらでも大きくできるので、$T(n) \leq cn$ が成り立つように、十分に大きな定数 c をとればよい。

このように定数 c の値は事前に決めておくのではなく、$T(n) \leq cn$ が成り立つように必要に応じて大きくして辻褄をあわせるという論法を使う。また $T(n)$ の条件も、n が十分に大きい時に成立してほしい条件

$$T(n) \leq T(\lceil n/5 \rceil) + T(7n/10 + 6) + an$$

をまず考え、この条件が成立つような n が十分に大きい場合の境界をあとで決める。最後に、あたかも初めから決めていたかのごとく、以下のように整理する。

$T(n) \leq cn$ if $n \leq 70$　（ただし c は不等式が成立つように十分大きくとる）

$T(n) \leq T(\lceil n/5 \rceil) + T(7n/10 + 6) + an$ if $n \geq 71$　（ただし $c \geq 71a$）

なお MIT Press 教科書では、 n が十分に大きい場合を $n \geq 140$ として、$c \geq 20a$ となるように定数 c を採用している。教科書には 140 を選択することは重要ではなく、$c \geq 10a(n/(n - 70))$　が満たされればよいことが書かれている。