Algorytmy i struktury danych

Lista zadań 2

Zadanie 1

Ile trzeba porównań, by znaleźć element x w nieuporządkowanej tablicy `t` o rozmiarze n. Oblicz wartość średnią i wariancję zakładając, że element x może znajdować się z jednakowym prawdopodobieństwem, pod dowolnym indeksem tablicy.

$$E(X) = \sum_{i=1}^{n} x_i p_i = \sum_{i=1}^{n} i \cdot \frac{1}{n} = \frac{\left(\frac{1}{n} + \frac{n}{n}\right) n}{2} = \frac{n+1}{2}$$

$$Var(X) = E(X^2) - E(X)^2 = \frac{n^2+1}{2} - \left(\frac{n+1}{2}\right)^2 = \frac{2n^2+2}{4} - \frac{n^2+2n+1}{4} = \frac{(n-1)^2}{4}$$

Zadanie 2

Bisekcja. Ile trzeba porównań, by znaleźć element x w posortowanej tablicy `t` o rozmiarze n. Podaj minimalną wartość gwarantującą sukces i strategię, jak to zrobić. Postaraj się podać wzór ogólny, który pozwoli wyliczyć dokładną wartość dla dowolnego n. Sprawdź go dla $n = 1, \ldots, 20$.

1. Oblicz środek przedziału.
2. Jeżeli wartość w środku przedziału jest równa x, to zakończ działanie algorytmu.
3. Jeżeli wartość w środku przedziału jest większa od x, to środek staje się lewym końcem przedziału, w przeciwnym wypadku prawym.

$$n = 2 \implies 3$$
$$n = 4 \implies 5$$
$$n = 8 \implies 7$$
$$n = 16 \implies 9$$
$$n = 20 \implies 9$$
$$2\lfloor \log_2(n) \rfloor + 1$$

Zadanie 3

Rozważ trzy wersje znajdowania maksimum w tablicy `int maks(int t[], int n)`. Ile porównań między elementami tablicy n-elementowej wykonuje każda z wersji? Ile pamięci wymaga każda z tych wersji? Uwzględnij fakt, że głębokość rekurencji ma wpływ na zużycie pamięci, ponieważ powstaje wiele kopii zmiennych lokalnych. Która wersja jest więc najlepsza?

(a) iteracyjna: `{int x = a[--n]; while(n--) if(t[n] > x) x = t[n]; return x;}`

$$n-1 \text{ porównań i 3 zmienne lokalne}$$

(b) rekurencyjnie oblicza maksimum $n-1$ elementów i porównuje z ostatnim elementem

$$n-1 \text{ porównań i } 2n \text{ zmiennych lokalnych}$$

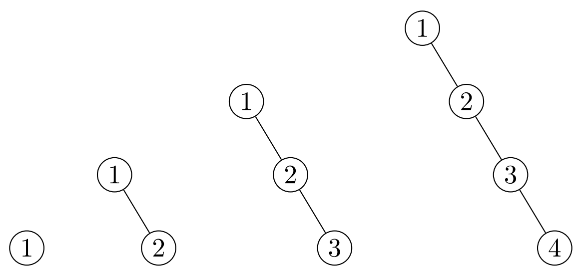
(c) dzieli tablicę na dwie części, rekurencyjnie znajduje ich maksima i wybiera większe z nich.

$$2(n-1)+1 = n-1 \text{ porównań i } 2(n-1)+2 = 2n \text{ zmiennych lokalnych}$$

Zadanie 4

Jakie drzewo powstanie po wstawieniu do pustego drzewa BST liczb od 1 do n w kolejności rosnącej? Jaka potem będzie głębokość drzewa? Ile porównań kluczy wykonano w trakcie tworzenia tego drzewa? Jaka jest złożoność w tego procesu w notacji O?

Uwaga: Element wstawiamy na pierwsze napotkane puste miejsce zaczynając od korze- nia. Jeśli miejsce jest zajęte, to gdy element jest mniejszy od klucza w węźle, idziemy do lewego poddrzewa, a gdy większy lub równy – do prawego poddrzewa.



Głębokość drzewa: n

$$\text{Ilość porównań: } \sum_{i=1}^{n} i-1 = \frac{n(n-1)}{2} = \frac{n^2-n}{2}$$

$$\text{Złożoność: } \frac{n^2-n}{2} = O(n^2)$$