Algorytmy i struktury danych

Lista 3

Zadanie 2

Krzysztof Juszczyk (307542)

21 kwietnia 2020

1. Rozwiązanie

Funkcję $f(n) = \lceil \frac{3}{2}n \rceil - 2$ możemy zdefiniować następująco:

$$f(n) = \begin{cases} 1 & \text{gdy n} = 1 \\ f(n-1) + 1 & \text{gdy n parzyste} \\ f(n-1) + 2 & \text{wpp.} \end{cases}$$

1.1. Obserwacja

Procedura $MaxMin$ rozpatruje n elementów. Dla n, takiego, że $2^{k-1} < n \leqslant 2^k$, możemy rozpatrzyć następujące przypadki:

1. $n = 2^k$
   Procedura $MaxMin$ na najniższych poziomach rekursji wykona dokładnie 2^{k-1} porównań między elementami na pozycjach parzystych i nieparzystych, wyznaczając w tych parach maximum i minimum. Na innych poziomach rekursji zostanie wykonane $2^k - 2$ porównania. W sumie wykonamy $\frac{3}{2}n - 2$ porównań.

2. $2^{k-1} + 2^{k-2} \leqslant n < 2^k$
   Dla n procedura $MaxMin$ na najniższym poziomie rekursji, w skrajnym prawym rozwinięciu drzewa zostanie, wykonane o 1 porównanie mniej niż dla $n + 1$ elementów. Liczba porównań na wyższych poziomach rekursji nie zmieni się.

3. $2^{k-1} < n < 2^{k-1} + 2^{k-2}$
   Dla n procedura $MaxMin$ względem $n + 1$ na najniższym poziomie rekursji, w skrajnym prawym rozwinięciu drzewa, zamiast wykonać 3 porównania dla 3 elementów wykona 1 porównanie dla 2 elementów. Zatem liczba porównań zmniejszy się o 2 względem wywołania $MaxMin$ na $n + 1$ elementach. Tutaj również liczba porównań na wyższych poziomach rekursji nie zmieni się.

1.2. Rozwiązanie

Pokazaliśmy już, że dla $n = 2^k$ elementów liczba porównań będzie równa $f(n)$. Znając definicje rekurencyjną $f(n)$ możemy zauważyć, że również dla $n = 2^k \pm 1$ wartość $f(n)$ będzie równa liczbie porównań procedury $MaxMin$.

Rozpatrując n w przedziale $2^{k-1} < n \leqslant 2^k$, zauważmy, że liczba porównań rośnie szybciej niż wartość funkcji $f(n)$ dla $n \leqslant 2^{k-1} + 2^{k-2}$. Gdy $n > 2^{k-1} + 2^{k-2}$ wtedy to funkcja $f(n)$ ma większy przyrost wartości. Zatem na przedziale $[2^{k-1}...2^k]$ największa różnica między liczbą porównań, a wartością $f(n)$ następuje dla $n = 2^{k-1} + 2^{k-2}$.

Co można zrobić aby zagwarantować ilość porównań równą $f(n)$? Gdy rozmiar rozpatrywanego zbioru S jest większy niż 2 to zamiast dzielić go na 2 równoliczne zbiory (z dokładnością do 1 elementu), możemy podzielić go na podzbiór S_1 o mocy 2^k i podzbiór S_2 zawierający resztę elementów. Oczywiście chcemy aby k było największe możliwe takie, że $2^k < n$. Wtedy na najniższym poziomie rekursji procedury $MaxMin$ będziemy wykonywać najmniejszą liczbę porównań. Liczba porównań na wyższych poziomach rekursji nie zmieni się.