**ASD – klasówka 1**

**22.11.2018**

**Zadanie 1 [7 punktów]**

Niech *n* będzie dodatnią liczbą całkowitą większą od 1. Dla dodatniej liczby całkowitej *k* < *n* powiemy, że ciąg liczb *a*[1],...,*a*[*n*] jest *k-*dobry, jeżeli dla każdej pary (*i,j*) takiej, że 1 ≤ *i* < *j* ≤ *n* oraz *a*[*i*] > *a*[*j*], mamy *j* ≤ *i* + *k*. Sortowaniem *k-*dobrym nazywamy takie porządkowanie elementów danego ciągu, w wyniku którego uzyskujemy ciąg *k*-dobry.

Zaproponuj algorytm sortowania *k*-dobrego działający w czasie O(*n*log (*n*/*k*)).

**Zadanie 2 [4 punkty]**

Wiadomo, że każdy algorytm wyznaczający przez porównania dwa elementy – największy i najmniejszy – w ciągu długości *n* = 2*k*, wykonuje w pesymistycznym przypadku co najmniej 3*k* – 2 porównania. Udowodnij powyższe dla *n* = 4 i zaproponuj optymalny algorytm w tym przypadku.

**Zadanie 3 [6 punktów]**

Elementy w tablicy *a*[1..*n*], *n* > 0, są rozmieszczone w porządku kopcowym typu MIN. Dla *k* > 0 tablicę *a* rozszerzono o *k* elementów *a*[*n*+1], …, *a*[*n*+*k*]. Zaproponuj algorytm, który w czasie O(*k* + log2*n*) zbuduje kopiec typu MIN na całej tablicy *a*[1..*n*+*k*].

Uwaga: rozpocznij swoje rozważania przy założeniu, że *n* = 2*h*-1, dla pewnego *h* > 0.

**Zadanie 4 [3 punkty]**

W tablicy *a*[1..*n*] dany jest ciąg jednocześnie 7- i 11-uporządkowany. Udowodnij, że algorytm InsertionSort sortuje *a* w czasie liniowym.

Uwaga: powiemy, że ciąg *a* jest *k*-uporządkowany, *k* > 0, gdy dla każdego *i* = 1, 2, …, *n* – *k*, *a*[*i*] ≤ *a*[*i*+*k*].

Uzasadnij poprawność swoich rozwiązań i przeprowadź analizę złożoności obliczeniowej zaproponowanych algorytmów.

**Bonusy:**

1. Udowodnij, że algorytm z zadania 1 jest optymalny.
2. W zadaniu 3 podaj algorytm działający w czasie O(*k* + log*n*).