AiSD, egzamin – 4 lutego, 2014 (z poprawkami)

1. (15p) Niech $f \prec g$ oznacza, że $f \in O(g)$ i $g \notin O(f)$ a $f \approx g$ oznacza, że $f \in O(g)$ i $g \in O(f)$. Uporządkuj następujące funkcje: $0.1n^2$, $2\sqrt{n} + n$, $n \log_2(n)$, $2n \log_5(n^2)$.

2. (15p) Oszacuj z dokładnością do O(.) czas działania algorytmu (wzgl. rozmiaru tablicy n = koniec - poczatek + 1).

```
Bla(int *t, int poczatek, int koniec){
  if(koniec - poczatek <= 3) return;
  for(i:=poczatek to i=koniec) print(t[i]);
 tmp:=(koniec - poczatek)/3;
  Bla(t,poczatek, poczatek + tmp);
  Bla(t, poczatek + tmp, poczatek + 2*tmp);
}
```

3. (15p) Sortowanie tablicy $t[1 \ldots N] = \{1, 2, 3, 4, 5, 6, 7\}$ algorytmem heapsort. Sterte utwórz w pętli **for(** `i:=N/2` **to** `1`**)** `downheap(t, i, N);` Na wierzchołku sterty umieść element największy. Podaj, jak wyglądać będzie sterta a następnie jak wyglądać będzie tablica po wstawianiu kolejnych elementów na swoje miejsce (w sumie wymaga to wypisania tablicy 7 razy, łącznie z ostatnią posortowaną).

4. (15p) Wypisz kolejne drzewa BST powstałe przez wykonanie następujących operacji na pustym drzewie: i(6), i(10), i(8), i(4), i(9), i(2), i(3), d(6), i(11), i(1), d(10). Przyjmij, że usuwając element, który nie jest liściem wpisujesz na jego miejsce odpowiedni element z jego prawego podrzewa.

5. (15p) Napisz algorytm `liscie(node *)` , który wywołany dla argumentu `tree` zwróci liczbę liści w drzewie, na które wskazuje `tree`. Przyjmij, że `node` jest strukturą postaci `struct node{int key; node *left; node *right;}`

6. (15p) Wyszukujemy z 8 elementów o kluczach: a, b, c, g, h ,x, y, z. Prawdopodobieństwa wyszukiwania elementu o danym kluczu są podane w nawiasach: a(0.1), b(0.05), c(0.3), g(0.1), h(0.05), x(0.2), y(0.1), z(0.1). Zbuduj optymalną listę nieuporządkowaną, która pozwoli zminimalizować koszt wyszukania jednego elementu (5p). Oblicz średni koszt wyszukania elementu na liście (czyli średnią liczbę odwiedzonych elementów listy) (5p). Możesz założyć, że nigdy nie są wyszukiwane elementy spoza listy.