AiSD, egzamin – 15.09.2015

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: $n\log_2(n)$, $n\log_{10}(n)$, $10^{\log_2(n)}$, $10^{\log_{10}(n)}$.

2. (15p) Wyznacz f taką, że czas działania algorytmu znajduje się w $\Theta(f)$:
```
Hoho(int *t, int poczatek, int koniec){int suma:=0;
  if((koniec - poczatek)<=2) return suma;
  for(j:=poczatek to j=koniec)
    for(k:=poczatek to k=koniec) suma:=suma+(t[j]* t[k]);
  polowa=poczatek+(koniec-poczatek+1)/2;
  return suma + Hoho(t, poczatek, polowa)+Hoho(t,polowa+1, koniec); }%Hoho
```

3. (15p) Przesortuj tablicę $t[1\ldots8] = \{7,3,6,9,11,1,18\}$ algorytmem sortowania przez kopcowanie (heap sort). Sterte utwórz w pętli:
for(`i:=N/2` **to** `1`**)** `downheap(t, i, N);`
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) Niech `fun(x)` to następująca funkcja:
```
int fun(x)
{
  if (x<=1) return 1;
  return fun(x-3) - fun(x-5);
}
```
Wypisz drzewo wywołań rekurencyjnych dla `fun`(9) oraz oblicz zwróconą wartość.

5. (15p) Napisz funkcję `IleParz(node *)`, która wywołana dla argumentu `head` zwróci ilość elementów zawierających parzystą wartość w polu key znajdujących się na liście, na którą wskazuje `head`. Przyjmij, że `node` jest strukturą postaci `struct node{int key; node *next;}`. Pamiętaj o przypadku gdy argument funkcji jest wskazaniem pustym (null) i przyjmij, że zwracana wartość wyniesie wtedy 0.

6. (15p) Prawdopodobieństwa wyszukiwania elementu o danym kluczu są podane w nawiasach: a(0.1), b(0.3), c(0.1), d(0.05), e(0.05), f(0.2), g(0.1), h(0.1). Na liście trzymamy tylko elementy a, b, d, e, f, h. Oblicz średni koszt wyszukania elementu (zakończonego powodzeniem lub nie) zdefiniowany jako liczba odwiedzonych na liście elementów dla listy uporządkowanej alfabetycznie oraz dla optymalnej listy nieuporządkowanej. Rozstrzygnij, która z tych struktur będzie efektywniejsza.