

Algorytmy i struktury danych 2009 lista 1

1. Ile co najmniej potrzeba porównań by posortować 2, 3, 4, 5, 6, 7 liczb?
2. Powiąż ilość porównań wykonywanych przez algorytm `insertion_sort` z ilością inwersji w ciągu wejściowym. Oblicz wartość oczekiwaną tej wielkości.
3. Dane jest struktura węzła listy pojedynczo łączonej:

```
struct lnode
{int k;
  lnode * next;
};
```

Zaimplementuj:

```
int first(lnode *l);
void insert(lnode*&l, int x);
lnode* find (lnode* l, int x);
bool remove (lnode*&l, int x);
```

4. Napisz funkcję odwracającą listę `void reverse(lnode*&l);`
5. Zależność:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Wykorzystaj do napisania rekurencyjnej funkcji `int Newton(int n, int k)`.
Oblicz złożoność czasową $T(n, k)$ swojej implementacji.

6. Wykorzystaj fakt, że:

$$\binom{n}{k} = n/1 * (n-1)/2 * (n-2)/3 * \dots * (n-k+1)/k$$

do napisania procedury `void Newtony(int n, int k, int Wyniki[n])`
wyznaczającej w czasie $O(n)$ wszystkie współczynniki Newtona:

$$\binom{n}{0}, \binom{n}{1}, \binom{n}{2}, \dots, \binom{n}{n}.$$