SPRAWDZIAN I

Imię i nazwisko:

Nr indeksu:

Nr grupy:

Uwaga! Sprawdzian jest testem wielokrotnego wyboru, gdzie wszystkie możliwe kombinacje odpowiedzi są dopuszczalne (tj. zarówno wszystkie odpowiedzi poprawne, część odpowiedzi poprawna jak i brak odpowiedzi poprawnych). Poprawne odpowiedzi należy zaznaczyć, z lewej strony kartki, symbolem "+". Natomiast symbol "-" jak i brak symbolu przy odpowiedzi oznacza odpowiedź niepoprawną. Pytanie jest uznane za poprawnie rozwiązane (tj. +1pkt) wtedy i tylko wtedy gdy wszystkie jego odpowiedzi zaznaczone są poprawnie. Życzymy powodzenia ...

1. Niech $f(n) = \sqrt{n} \lg n!$, wtedy prawdą jest, że:
   (a) **[+]** $f(n) = O(n^2)$,
   (b) **[+]** $f(n) = \Omega(n\sqrt{n})$,
   (c) **[+]** $f(n) = \Theta\left(2^{\lg \sqrt{n}} n \lg n\right)$.

2. Rozważmy funkcję $f : \mathbb{N} \to \mathbb{R}^+ \cup \{0\}$ postaci $f(n) = 2^n$, wtedy:
   (a) **[+]** $f(n) = \Theta(c \cdot f(n) + 1)$, gdzie c jest pewną dodatnią stałą,
   (b) **[–]** $f(n) = O\left(\frac{1}{n!} \cdot f(n)\right)$,
   (c) **[–]** $f(n) = \Omega\left(f(n)^2\right)$.

3. Które z poniższych zdań jest prawdziwe:
   (a) **[+]** jeżeli $f(n) = O(n)$ i $g(n) = O(n)$, to $f(n) + g(n) = O(n^2)$,
   (b) **[+]** jeżeli $f(n) = O(n^2)$ i $g(n) = O(n^2)$, to $f(n) + g(n) = O(n^2)$,
   (c) **[–]** jeżeli $f(n) = \Omega(n)$ i $g(n) = \Omega(n)$, to $f(n) + g(n) = \Omega(n^2)$.

4. Załóżmy, że złożoność czasową pewnego algorytmu A określa funkcja $T(A, n) = \sqrt{n}$, gdzie n jest rozmiarem danych wejściowych. Komputer K wykonuje rozważany algorytm dla danych rozmiaru 36 w ciągu 12 sekund, tj. $T_K(A, 36) = 12$. Stąd:
   (a) **[–]** $T_K(A, 49) = 16$,
   (b) **[–]** $T_K(A, 49) = 18$,
   (c) **[+]** w ciągu 100 sekund komputer K wykona rozważany algorytm dla danych wejściowych rozmiaru co najwyżej 2500.

5. Rozważmy następujący algorytm

```
void Algorytm(int n) {
   Alg1(n);
   for (i=0;i<n*n;i++) {
      Alg2(n);
   }
}
```

   gdzie Alg_1 oraz Alg_2 są algorytmami o złożoności czasowej odpowiednio $T(Alg_1 n) = O(n \lg n!)$ oraz $A(Alg_2, n) = \Theta(n)$, $W(Alg_2, n) = \Theta(n^2)$, stąd:
   (a) **[–]** $T(Algorytm, n) = \Theta(n^2 \lg n!)$,

(b) [+] $A\,(Algorytm, n) = O\left(n^2 \lg n!\right)$,

(c) [+] $W\,(Algorytm, n) = \Omega\left(n^2 \lg n!\right)$.

6. Rozważmy następujący algorytm

```
int Cos(int n) { // wp:  n∈ ℕ
   int i=10;
   while (i≥0) i=i+1;
   return n; // wk:  n∈ ℕ
}
```

wtedy:

(a) [–] program Cos jest całkowicie poprawny w strukturze liczb naturalnych,

(b) [+] program Cos jest częściowo poprawny w strukturze liczb naturalnych,

(c) [+] program Cos jest całkowicie poprawny w strukturze liczb naturalnych przy założeniu, że operator dodawania zdefiniujemy jak odejmowanie, tj. $+ =_{def} -$.

7. Rozważmy następujący algorytm

```
int Cos(int n, int k) {
   int i=k, wynik=1;
   while (i≤n) {
      i=i*k;
      wynik=wynik+1;
   }
   return wynik; // wk:  wynik=log_k n+1
}
```

wtedy:

(a) [–] program Cos jest częściowo poprawny dla warunku początkowego $k, n \in \mathbb{N}$,

(b) [+] program Cos jest częściowo poprawny dla warunku początkowego $n = k^c$, dla $c \in \mathbb{N}^+$ i $k \in \mathbb{N} \setminus \{0, 1\}$,

(c) [+] program Cos jest całkowicie poprawny dla warunku początkowego $n = k^c$, dla $c \in \mathbb{N} \setminus \{0, 1, 2, \ldots, k\}$ i $k \in \mathbb{N} \setminus \{0, 1\}$.

8. Rozważmy następujący algorytm

```
int Cos(int n) { // wp:  n∈ ℕ
   int i=0, s=0;
   while (i<n) {
      i=i+1;
      s=s+i;
   }
   return s;
}
```

wtedy:

(a) [+] niezmiennikiem pętli w programie Cos jest formuła $s = \frac{i(i+1)}{2}$,

(b) [–] niezmiennikiem pętli w programie Cos jest formuła $s = \frac{i(i-1)}{2}$,

(c) [+] niezmiennikiem pętli w programie Cos jest formuła $i \in \mathbb{N}$, a warunkiem końcowym $s = \sum_{i=0}^{n} i$.

9. Które ze zdań jest prawdziwe:

(a) [–] sprawdzenie, czy dany element należy do nieuporządkowanego uniwersum rozmiaru n wymaga $O\left(\sqrt{n}\right)$ porównań,

(b) [+] sprawdzenie, czy dany element należy do nieuporządkowanego uniwersum rozmiaru $\sqrt{n}$ wymaga $O\,(n)$ porównań,

(c) [+] koszt czasowy sekwencyjnego algorytmu wyszukania elementu minimalnego w nieuporządkowanym uniwersum rozmiaru 10^6 wynosi $10^6 - 1$.

10. Rozważmy algorytm „turniej” dla danych rozmiaru $n = 2^k$, gdzie $k \in \mathbb{N}^+$. Które z poniższych stwierdzeń jest zawsze spełnione:

    (a) [–] koszt budowy drzewa turnieju wynosi dokładnie $n + 1$ porównań,

    (b) [–] element 2-gi co do wielkości „pojedynkował się” dokładnie z $\lg n - 1$ elementami,

    (c) [+] element 1-szy co do wielkości „pojedynkował się” dokładnie z $\lg n$ elementami.

11. Rozważmy iteracyjny algorytm dla problemu min-max i danych rozmiaru $n = 2^k$, gdzie $k \in \mathbb{N}^+$, wtedy:

    (a) [+] algorytm ten jest optymalnym rozwiązaniem dla rozważanego problemu,

    (b) [+] złożoność czasową algorytmu można oszacować przez $\frac{3}{2}n \pm c$, gdzie $c \leq 3$,

    (c) [+] złożoność pamięciowa algorytmu jest rzędu $O(\lg n)$.

12. Które ze zdań jest prawdziwe:

    (a) [–] sprawdzenie algorytmem BinSearch, czy dany element należy do nieuporządkowanego uniwersum rozmiaru n wymaga $O(1)$ porównań,

    (b) [+] sprawdzenie algorytmem BinSearch, czy dany element należy do uporządkowanego uniwersum rozmiaru n wymaga $O(\lg n)$ porównań,

    (c) [+] koszt czasowy algorytmu BinSearch dla poprawnych danych rozmiaru 1111 wynosi co najwyżej 12 porównań.

13. Załóżmy, że pewien algorytm Alg dla danych wejściowych rozmiaru n składa się z dwóch części:

    - $\sqrt{n}$-krotne wyszukanie elementu minimalnego metodą sekwencyjną,
    - $\lg n$-krotne wyszukanie elementu minimalnego algorytmem Hoare'a.

    Które z oszacowań jest poprawne:

    (a) [+] $A(Alg, n) = \Omega(n\sqrt{n})$,

    (b) [–] $W(Alg, n) = O(n\sqrt{n}\lg n)$,

    (c) [+] $S(Alg, n) = \Theta(1)$.

14. Który z poniższych ciągów jest poprawnym rezultatem wykonania procedury Split dla danych wejściowych

    $$4, 7, 9, 11, 3, 5, 2,$$

    (a) [–] 4,2,3,11,9,5,7,

    (b) [–] 2,3,4,5,7,9,11,

    (c) [+] 3,2,4,11,9,5,7.

15. Rozważmy wyszukiwanie elementu n-tego co do wielkości, w n-elementowej uporządkowanej rosnąco tablicy wejściowej, przy zastosowaniu algorytmu Hoare'a z procedurą podziału zgodną z metodą Partition, wtedy:

    (a) [–] złożoność czasową rozwiązania w tym przypadku szacujemy przez $O(1)$,

    (b) [–] złożoność czasową rozwiązania w tym przypadku szacujemy przez $O(n)$,

    (c) [+] złożoność czasową rozwiązania w tym przypadku szacujemy przez $O(n^2)$.

16. Prowadzący zajęcia ćwiczeniowe z ASD jest:

    (a) leworęczny,

    (b) praworęczny,

    (c) nie wiem, ale z dokładnością do notacji $\Theta(1)$ używa jednej ręki.