

Wstęp do Informatyki 2023/2024

Lista 7

Instytut Informatyki, Uniwersytet Wrocławski

28 listopada, 1 i 13 grudnia 2023

Uwagi: Uzasadnienia poprawności funkcji/algorytmów w Waszych rozwiązaniach *nie muszą być tak formalne* jak (niektóre) dowody poprawności na wykładzie 7. Należy jednak precyzyjnie formułować *niezmienniki pętli* i uzasadniać, że ich spełnienie (wraz z zaprzeczeniem warunku wejścia do pętli) pomaga dowieść pożądane własności.

1. [1] Twierdzymy, że poniższy program P jest częściowo poprawny względem następującej formalnej specyfikacji:

$$\{n > 0\} P \{ \text{res} = |\{j : 0 \leq j < n \wedge a[j] = x\}| \}$$

gdzie $|A|$ oznacza liczbę elementów zbioru A . (Zauważ, że podany warunek końcowy można sformułować opisowo: ‘res jest równe liczbie elementów równych x w ciągu $a[0], a[1], \dots, a[n-1]$ ’.)

Sformułuj i udowodnij niezmiennik pętli, który pozwala formalnie wykazać częściową poprawność P względem tej specyfikacji.

Program P :

```
{
  int i = 0, res = 0;
  while (i != n) {
    if (a[i] == x)
      res = res + 1;
    i = i + 1;
  }
}
```

2. [1] Uzasadnij, że wartość zwracana przez poniższą funkcję dla naturalnego $b > 0$ jest równa a^b . W swoim rozwiązaniu sformułuj niezmiennik pętli, uzasadnij jego poprawność.

```

int potega(int a, int b) {
    int n = a, k = b, res = 1;
    while (k != 0) {
        if (k % 2 != 0) {
            res = res * n;
            k = k - 1;
        }
        else {
            n = n * n;
            k = k / 2;
        }
    }
    return res;
}

```

Wskazówka: Formułując niezmiennik zajrzyj do dowodu poprawności algorytmu potęgowania podanego na wykładzie 7.

3. [1] Wykonaj poniższe polecenia:

(a) Uzupełnij warunek końcowy dla specyfikacji funkcji **znajdz**:

Warunek początkowy:

n, x — liczby naturalne,

a — tablica liczb naturalnych, taka że $a[0] \leq a[1] \leq \dots \leq a[n-1]$

Warunek końcowy:

- $res = -1$ gdy
- $res \in [0, n-1]$ takie, że
gdy

(b) Uzasadnij (intuicyjnie) zgodność funkcji ze specyfikacją z punktu (a).

(c) Sformułuj poprawny niezmiennik pętli, uzasadnij jego przydatność w wykazaniu częściowej poprawności całej funkcji względem specyfikacji podanej w punkcie (b).

```

int znajdz(int n, int a[], int x) {
    int b = 0, e = n - 1, s, res = -1;
    while (b <= e) {
        s = (b + e) / 2;    // dzielenie całkowite
        if (a[s] == x) {
            res = s;
            b = e + 1;
        }
        else if (x < a[s])
            e = s - 1;
        else
            b = s + 1;
    }
    return res;
}

```

4. [1] Uzasadnij, że

(a) Funkcja **potega** spełnia własność stopu (przyjmujemy, że $b > 0$ to liczba naturalna dodatnia).

- (b) Funkcja **znajdz** spełnia własność stopu.
5. [1] Ustal jakie jest działanie poniższej funkcji **max**, zakładając, że w tablicy *a* znajdują się liczby całkowite (w tym mogą być również liczby ujemne). Następnie sformułuj niezmiennik pętli, który pomoże uzasadnić Twoją odpowiedź.

```
int max(int n, int a[]) {  
    int i = 0, s = 0, ms = a[0];  
    while (i < n) {  
        s = s + a[i];  
        if (s > ms)  
            ms = s;  
        i++;  
    }  
    return ms;  
}
```

6. [1] Zakładamy, że w tablicy *a* umieszczono **tylko** liczby **nieujemne**. Dla poniższej funkcji **segment**:
- (a) Ustal jaki jest efekt jej działania gdy $w > 0$. Odpowiedź uzasadnij. (Twoje uzasadnienie nie musi być formalne.)
- (b) Uzasadnij, że funkcja **segment** spełnia własność stopu dla $w > 0$.

```
int segment(int n, int a[], int w) {  
    int l = 0, p = 0, s = 0;  
    while (s != w && p < n - 1) {    // 'is' is 'and'  
        if (s < w) {  
            s = s + a[p];  
            p++;  
        }  
        else {  
            s = s - a[l];  
            l++;  
        }  
    }  
    while (s > w) {  
        s = s - a[l];  
        l++;  
    }  
    if (s == w)  
        return 1;  
    else  
        return 0;  
}
```

Zadania dodatkowe, niedeklarowane (nie wliczają się do puli punktów do zdobycia na ćwiczeniach, punktacja została podana tylko jako informacja o trudności zadań wg wykładowcy):

7. [0,5] Uzasadnij, że wszystkie typy pętli w Twoim ulubionym języku programowania można zaimplementować wykorzystując jedynie pętlę **while**.