

Wstęp do informatyki

Lista 7

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] Twierdzimy, że poniższy program P jest częściowo poprawny względem następującej formalnej specyfikacji

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

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

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 poti(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 `znajdzI`:

Warunek początkowy: n, x – liczby naturalne,

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

Warunek końcowy:

- $\text{res} = -1$ gdy
- res z przedziału $[0, n-1]$ takie, że
gdy

b) (intuicyjnie) uzasadnij 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 znajdzI(int n, int a[], int x)
{ int b,e,s,res;
  b = 0; e = n - 1;
  res = -1;
  while (b<=e){
    s = (b+e)/2;
    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 `potI` spełnia własność stopu (przyjmujemy, że b to liczba naturalna dodatnia).
- b) Funkcja `znajdzI` spełnia własność stopu.

5. [1] Ustal jakie jest działanie poniższej funkcji. Następnie sformułuj niezmiennik pętli, który pomoże uzasadnić Twoją odpowiedź.

```
int maxS(int n, int a[])
{ int ms, i;
  i=1; ms=a[0]; s=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 spełnia własność stopu dla $w > 0$.

```
int segment(int n, int a[], int w)
{ int l, p, s;
  l=p=0;
  s=0;
  while (s != w && p < n) { // && to spójnik 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, nieobowiązkowe (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*.