Homework 1 (due September 29)

Problem 1 (0.1 point). (a) The sum of arithmetic progression $a_k, a_{k+1}, \dots, a_l$ (where $a_i = a_k + (i-k) \cdot d$, $i = k+1, \dots, l$) may be expressed by formula

$$\sum_{j=k}^{l} a_j = \frac{a_k + a_l}{2}(l - k + 1).$$

The sum of geometric progression $b_1, b_2, b_3, \dots, b_k$ (where $b_i = b_1 \cdot q^{i-1}$, $i = 2, \dots, k$ and $q \neq 1$) may be expressed by formula

$$\sum_{i=1}^{k} b_i = b_1 \frac{1 - q^k}{1 - q}.$$

Using these formulas as well as formula

$$1^2 + 2^2 + 3^2 + \dots + n^2 = \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6},$$

which may be easily proved by mathematical induction, find the sum $f(n) = \sum_{k=u(n)}^{v(n)} g(k)$.

(b) Find asymptotics of $f(n)$, i.e. constants a and b such that $f(n) \sim an^b$, when $n \to \infty$. If $f(n)$ grows exponentially, then find constants a and b such that $f(n) \sim ab^n$.

Remark. $f(n) \sim g(n)$ ("f is asimptotically equal to g") if

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 1.$$

Exercises

1. $f(n) = 2^2 + 4^2 + 6^2 + \dots + (2n)^2$;
2. $f(n) = \sum_{k=2}^{n-1} k^2 - (\sum_{k=3}^{n} k)^2$;
3. $f(n) = 1 \cdot 2 + 2 \cdot 3 + 3 \cdot 4 + \dots + (n-1) \cdot n$;
4. $f(n) = \sum_{k=1}^{n} (k+1)^2$;
5. $f(n) = 1^2 - 2^2 + 3^2 - 4^2 + \dots + (-1)^{n-1} n^2$, where n is an odd number;
6. $f(n) = \sum_{k=1}^{[n/2]} k^2$, where $[x]$ is an integer part of number x;
7. $f(n) = 1^2 - 1 + 2^2 + 2 + 3^2 - 3 + \dots + n^2 + (-1)^n n$;
8. $f(n) = \sum_{k=-n}^{n} (k^2 + k)$;
9. $f(n) = 1 + 2 - 3 + 4 + 5 - 6 + \dots + (n-2) + (n-1) - n$, where n is a multiplier of 3;

10. $f(n) = \sum_{k=1}^{n-1} k(k+1)$;

11. $f(n) = n + \frac{n}{2} + \frac{n}{4} + \cdots + \frac{n}{n/2} + 1$, where $n = 2^k$;

12. $f(n) = \sum_{k=1}^{n} (3^k - k^2)$;

13. $f(n) = 1 \cdot 3 + 2 \cdot 4 + 3 \cdot 5 + \cdots + (n-2) \cdot n$, where $n \geq 3$;

14. $f(n) = 1 + 2^1 + 2 + 2^2 + 3 + 2^3 + \cdots + n + 2^n$;

15. $f(n) = \sum_{k=1}^{2n} (-1)^k k^2$;

16. $f(n) = 2 - 1 + 2^2 - 2 + 2^3 - 3 + \cdots + 2^n - n$;

17. $f(n) = \sum_{k=1}^{n} (k^2 - 2k)$;

18. $f(n) = \sum_{k=1}^{n} (2^k + k^2)$;

19. $f(n) = \sum_{k=1}^{n} \frac{2^k - 1}{3^{k-1}}$;

20. $f(n) = n - \frac{n}{2} + \frac{n}{4} - \frac{n}{8} + \cdots + (-1)^k \frac{n}{2^k}$, where $n = 2^k$;

21. $f(n) = \sum_{k=-n}^{n} (2^k - k^2)$;

22. $f(n) = \sum_{k=1}^{n-1} (\frac{2^{k+1}}{3^k} + 1)$;

23. $f(n) = \sum_{k=1}^{n} (k-1)(k+1)$;

24. $f(n) = \sum_{k=1}^{n} [\frac{k}{2}]$, where $[x]$ is an integer part of number x;

25. $f(n) = \sum_{k=0}^{n-1} \frac{2^k + 1}{3^{k+1}}$;

26. $f(n) = \sum_{k=1}^{n-1} (2^k - k - 1)$;

27. $f(n) = 3 - 2 + 3^2 - 2^2 + \cdots + (3^n - 2^n)$;

28. $f(n) = \sum_{k=0}^{n/2} (2^k + k)$, where n is an even number;

29. $f(n) = \sum_{k=1}^{n} [\frac{k^2}{2}]$, where $[x]$ is an integer part of number x and n is an even number;

30. $f(n) = \sum_{k=1}^{[n/2]} (-1)^k 2^k$, where $[x]$ is an integer part of number x.

Problem 2 (0.3 point). Let us consider program code fragment depending on parameter n.

(a) Supposing that each operation (assignment, arithmetical, comparison etc.) has weight 1, find an exact number of operations $L(n)$. Number of operations should be counted for the worst case of initial data (maximum taken on all possible data of size n).

(b) Find asymptotics of $L(n)$, i.e. constants a and b such that $L(n) \sim an^b$, when $n \to \infty$.

(c) A small constant c is given. Indicate an example of initial data that requires exactly $L(c)$ steps and enumerate all these steps.

(d) Find order of growth for the program execution time $T(n)$, i.e., find a constant d such that $T(n) = \Theta(n^d)$, when $n \to \infty$. Counting time $T(n)$ we consider that execution of different operations (assignment, arithmetical, comparison etc.) takes different time: operation of type i requires time c_i.

Indications

1. $f(n) = \mathrm{O}(g(n))$ (or $f(n) \preccurlyeq g(n)$) (we say that "f has no higher asymptotical growth order than g") if $\exists N \in \mathbb{N}$ and $\exists c > 0$: $f(n) \leqslant cg(n)\ \forall n \geq N$;

2. $f(n) = \Theta(g(n))$ (or $f(n) \asymp g(n)$) (we say that "f and g have the same asymptotical growth order") if $f(n) = \mathrm{O}(g(n))$ and $g(n) = \mathrm{O}(f(n))$.

3. Counting steps we consider that both assignment and arithmetical operation require 1 step, i.e., instruction $a := 1$ takes 1 step and instruction $a := b+c$ also takes 1 step. However, instruction $a := b + c - d$ takes 2 steps. Instruction $A[i+j] := b+c$ also takes 2 steps: (1) we count index value $k = i + j$, and (2) we assign to $A[k]$ value $b + c$.

4. Execution of cycle **for** of length k, i.e. instruction **for** $i := 1$ **to** k **do** requires $2(k + 1)$ steps because each time adding $i := i + 1$ and comparison $i \leq k$? are executed. At the end of cycle once more adding $i := k+1$ and comparison $k+1 \leq k$? will be executed but the body of cycle will not be executed anymore.

Exercises

1. An array of integer numbers $A[1 : n]$ is given; $c = 2$.

   for $i := 1$ **to** n **do**

    $B[i] := 0$

    for $j := i$ **to** n **do**

     $B[i] := B[i] + A[j]$

    if $B[i] < A[i]$ **then** $B[i] := 0$

2. An array of integer numbers $A[1:n]$ is given; $c = 2$.

   for $j := 1$ **to** n **do**

   $\quad C[j] := 0$

   $\quad i := n$

   $\quad$ **while** $i \geq j$ **do**

   $\quad\quad C[j] := C[j] + A[i]$

   $\quad\quad i := i - 1$

   $\quad$ **if** $C[j] < 0$ **then** $C[j] := 0$

3. An array of integer numbers $A[1:n]$ is given; $c = 2$.

   for $j := 1$ **to** n **do**

   $\quad C[j] := 0$

   $\quad$ **for** $i := j + 1$ **to** n **do**

   $\quad\quad C[j] := C[j] + A[i]$

   $\quad$ **if** $C[j] < 0$ **then** $C[j] := -C[j]$

4. An array of integer numbers $A[1:n]$ is given; $c = 2$.

   for $j := 1$ **to** n **do**

   $\quad B[j] := 0$

   $\quad k := j - 1$

   $\quad$ **for** $i := 1$ **to** k **do**

   $\quad\quad B[j] := B[j] + A[i]$

   $\quad$ **if** $B[j] < 0$ **then** $B[j] := 0$

5. An array of integer numbers $A[1:n]$ is given; $c = 2$.

   for $j := 1$ **to** n **do**

   $\quad C[j] := 0$

   $\quad i := 1$

   $\quad k := n/2$

   $\quad$ **while** $i \leq k$ **do**

   $\quad\quad C[j] := C[j] + A[i]$

   $\quad\quad i := i + 1$

   $\quad$ **if** $C[j] < 0$ **then** $C[j] := 0$

6. An array of integer numbers $A[1:n]$ is given; $c = 2$.

   $i := 1$

   while $i \leq n$ **do**

   $B[i] := 0$

   for $j := i$ **to** n **do**

   $B[i] := B[i] + A[j]$

   if $B[i] < A[i]$ **then** $B[i] := 0$

   $i := i + 1$

7. An array of integer numbers $A[1:n]$ is given; $c = 2$.

   $j := 1$

   while $j \leq n$ **do**

   $C[j] := 0$

   $i := n$

   while $i \geq j$ **do**

   $C[j] := C[j] + A[i]$

   $i := i - 1$

   if $C[j] < 0$ **then** $C[j] := 0$

   $j := j + 1$

8. An array of integer numbers $A[1:n]$ is given; $c = 2$.

   $j := 1$

   while $j \leq n$ **do**

   $C[j] := 0$

   for $i := j + 1$ **to** n **do**

   $C[j] := C[j] + A[i]$

   if $C[j] < 0$ **then** $C[j] := -C[j]$

   $j := j + 1$

9. An array of integer numbers $A[1:n]$ is given; $c = 2$.

   $j := 1$

   while $j \leq n$ **do**

   $B[j] := 0$

   $k := j - 1$

   for $i := 1$ **to** k **do**

```
        B[j] := B[j] * A[i]
    if B[j] < 0 then B[j] := 0
    j := j + 1
```

10. An array of integer numbers $A[1:n]$ is given; $c = 2$.

```
j := 1
while j ≤ n do
    C[j] := 0
    i := 1
    k := n/2
    while i ≤ k do
        C[j] := C[j] * A[i]
        i := i + 1
    if C[j] < 0 then C[j] := 0
    j := j + 1
```

11. An array of integer numbers $A[1:n]$ is given; $c = 2$.

```
m := n − 1
for i := 1 to m do
    min := A[i]
    k := i
    for j := i + 1 to n do
        if A[j] < min then
            min := A[j]
            k := j
    A[k] := A[i]
    A[i] := min
```

12. An array of real numbers $A[0:n]$ and real number $z \in \mathbb{R}$ are given; $c = 1$.

```
S := 0
for i := 0 to n do
    d := A[i]
    for j := 1 to i do
        d := d * z
    S := S + d
```

13. An array of integer numbers $A[1:n]$ is given; $c = 2$.

$i := 1$

$m := n - 1$

while $i < n$ **do**

  for $j := m$ **step** -1 **to** i **do**

    if $A[j] > A[j+1]$ **then**

      $key := A[j+1]$

      $A[j+1] := A[j]$

      $A[j] := key$

  $i := i + 1$

14. An array of integer numbers $A[1:n]$ is given; $c = 2$.

for $i := 1$ **to** n **do**

  $S[i] := 0$

  for $j := 1$ **to** n **do**

    if $j \leq (n+1)/2$ **then** $S[i] := S[i] + A[j] * A[n-j]$

  $S[i] := S[i] * i$

15. An array of integer numbers $A[1:n]$ is given; $c = 2$.

$j := 1$

while $j < n$ **do**

  $\min := A[j]$

  $l := j$

  for $i := j + 1$ **to** n **do**

    if $A[i] < \min$ **then**

      $\min := A[i]$

      $l := i$

  $A[l] := A[j]$

  $A[j] := \min$

  $j := j + 1$

16. An array of real numbers $A[0:n]$ and real number $z \in \mathbb{R}$ are given; $c = 1$.

$T := 0$

for $j := 0$ **to** n **do**

  $d := A[j]$

```
i := 1
while i ≤ j do
    d := d * z
    i := i + 1
T := T + d
```

17. An array of integer numbers $A[1 : n]$ is given; $c = 2$.

```
j := 1
while j ≤ n do
    i := n − 1
    while i ≥ j do
        if A[i] > A[i + 1] then
            key := A[i]
            A[i] := A[i + 1]
            A[i + 1] := key
        i := i − 1
    j := j + 1
```

18. An array of integer numbers $A[1 : n]$ is given; $c = 2$.

```
j := 1
while j ≤ n do
    S[j] := 0
    if A[j] > 0 then l := 2 else l := 3
    for i := 1 step l to n do
        S[j] := S[j] + A[j] * A[i]
    j := j + 1
```

19. An array of integer numbers $A[1 : n]$ is given; $c = 2$.

```
for i := 1 to n do
    T[i] := 0
    l := 2 * i
    for j := l to n do
        S[i] := S[i] + A[j]
```

20. An array of integer numbers $A[1:n]$ is given; $c = 2$.

$m := n - 1$
for $j := 1$ **to** m **do**
    $\min := A[j]$
    $l := j$
    $i := n$
    while $i > j$ **do**
        if $A[i] < \min$ **then**
            $\min := A[i]$
            $l := i$
            $i := i - 1$
    $A[l] := A[j]$
    $A[j] := \min$

21. An array of real numbers $A[0:n]$ and real number $z \in \mathbb{R}$ are given; $c = 1$.

$S := 0$
$i := 0$
while $i \leq n$ **do**
    $e := A[i]$
    $j := 1$
    while $j \leq i$ **do**
        $e := e * z$
        $j := j + 1$
    $S := S + e$
    $i := i + 1$

22. An array of integer numbers $A[1:n]$ is given; $c = 2$.

$m := n - 1$
for $j := 1$ **to** m **do**
    for $i := m$ **step** -1 **to** j **do**
        if $A[i] > A[i+1]$ **then**
            $key := A[i]$
            $A[i] := A[i+1]$
            $A[i+1] := key$

23. An array of integer numbers $A[1:n]$ is given; $c = 2$.

$j := 1$

while $j \leq n$ **do**

    $S[j] := 0$

    $k := 2 * j$

    for $i := k$ **to** n **do**

        $S[j] := S[j] + A[i]$

    $j := j + 1$

24. An array of integer numbers $A[1:n]$ is given; $c = 2$.

for $i := 1$ **to** n **do**

    $S[i] := 0$

    $k := 1$

    if $A[i] < 0$ **then** $k := 2$

    for $j := i$ **step** k **to** n **do**

        $S[i] := S[i] + A[i] * A[j]$

25. An array of integer numbers $A[1:n]$ is given; $c = 2$.

$m := n - 1$

for $j := 1$ **to** m **do**

    $\min := A[j]$

    $k := j$

    $i := j + 1$

    while $i \leq n$ **do**

        if $A[i] < \min$ **then**

            $\min := A[i]$

            $k := i$

            $i := i + 1$

    $A[k] := A[j]$

    $A[j] := \min$

26. An array of real numbers $A[0:n]$ and real number $z \in \mathbb{R}$ are given; $c = 1$.

$j := 0$

$m := n + 1$

$T := 0$

while $j < m$ **do**

  $e := A[j]$

  for $i := 1$ **to** j **do**

    $e := e * z$

  $T := T + e$

  $j := j + 1$

27. An array of integer numbers $A[1 : n]$ is given; $c = 2$.

$m := n - 1$

for $i := 1$ **to** m **do**

  $j := m$

  while $j \geq i$ **do**

    if $A[j] > A[j + 1]$ **then**

      $key := A[j]$

      $A[j] := A[j + 1]$

      $A[j + 1] := key$

    $j := j - 1$

28. An array of integer numbers $A[1 : n]$ is given; $c = 2$.

for $i := 1$ **to** n **do**

  $S[i] := 0$

  if $A[i] > 0$ **then** $k := 2$ **else** $k := 3$

  for $j := 1$ **step** k **to** n **do**

    $S[i] := S[i] + A[j] * A[i]$

29. An array of integer numbers $A[1 : n]$ is given; $c = 2$.

$j := 1$

while $j \leq n$ **do**

  $S[j] := 0$

  $l := 1$

  if $A[j] > 0$ **then** $l := 2$

  for $i := j$ **step** l **to** n **do**

    $S[j] := S[j] + A[j] * A[i]$

  $j := j + 1$

30. An array of integer numbers $A[1:n]$ is given (where n is even); $c = 2$.

for $j := 1$ **to** n **do**

  $S[j] := 0$

for $j := n$ **step** -2 **to** 1 **do**

  $k := j/2$

  for $i := 1$ **to** k **do**

    $S[j] := S[j] + A[i]$