



ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
ТАМБОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ



*Доклад на тему:*

# Разработка рекурсивного алгоритма решения олимпиадной задачи по информатике на переливания



*Автор: Кулаков Юрий  
Владимирович,  
кандидат технических наук,  
доцент*

Тамбов 2022

# \* Актуальность работы

Насущная необходимость накопления теоретического и практического опыта разработки алгоритмов решения олимпиадных задач по информатике. Пропаганда научных знаний и подготовка учащихся к всероссийским и международным олимпиадам по информатике и программированию.

# \* Цель и задачи исследования

**Цель исследования:** Построение рекурсивного алгоритма решения олимпиадной задачи по информатике «Переливания».

**Задачи исследования:**

- 1) Изучить постановку олимпиадной задачи «Переливания».
- 2) Провести первичный анализ процесса переливаний с целью освобождения одного из трёх сосудов.
- 3) Спроектировать алгоритм решения олимпиадной задачи «Переливания».

# \* Объект, предмет и практическая значимость

**Объект исследования:** Олимпиадная задача по информатике и программированию «Переливания».

**Предмет исследования:** Алгоритм решения олимпиадной задачи по информатике и программированию «Переливания».

**Практическая значимость:** Приобретение знаний, умений и навыков разработки алгоритмов решения олимпиадных задач по информатике.

# \* Постановка олимпиадной задачи

Есть три сосуда с водой. В одном из них  $A$  миллилитров воды, в другом -  $B$  миллилитров, в третьем -  $C$ . Разрешается следующая операция. Можно перелить воду из одного сосуда в другой так, чтобы в том сосуде, в который мы переливаем, количество воды после переливания было в два раза больше, чем до переливания. То есть, если до переливания в сосудах было  $A$ ,  $B$  и  $C$  миллилитров соответственно, и мы переливаем, например, из второго сосуда в третий, то после переливания в сосудах должно оказаться  $A$ ,  $B - C$ ,  $2C$  миллилитров соответственно (такое переливание можно делать только при условии, когда  $B \geq C$ ). Эту операцию можно повторять не более 10000 раз.

Напишите программу, которая определит, можем ли мы в результате освободить один из сосудов.

# \* Первичный анализ олимпиадной задачи

Схематичное переливание из второго сосуда в третий

$$(A, B, C) -23-\> (A, B - C, 2C). \quad (1)$$

Первый пример:  $A = 1$  мл,  $B = 2$  мл и  $C = 10$  мл.

Пути переливаний (длины  $n = 2$ ):

$$(1, 2, 10) -31-\> (2, 2, 9) -12-\> (0, 4, 9) \quad (2)$$

или

$$(1, 2, 10) -31-\> (2, 2, 9) -21-\> (4, 0, 9). \quad (3)$$

Второй пример:  $A = 0$  мл,  $B = 1$  мл,  $C = 0$  мл.

Тривиальный путь (длины  $n = 0$ ):

$$(0, 1, 0). \quad (4)$$

# \* Варианты изначальной наполненности сосудов

1)  $A = 8, B = 3, C = 11;$

2)  $A = 10, B = 2, C = 6;$

3)  $A = 11, B = 14, C = 3;$

4)  $A = 4, B = 7, C = 2;$

5)  $A = 14, B = 5, C = 8;$

6)  $A = 5, B = 4, C = 6;$

7)  $A = 7, B = 6, C = 12.$

# \* Основной алгоритм решения задачи

- 1) Начало алгоритма.
- 2) Ввести значения наполненности сосудов  $A$ ,  $B$ ,  $C$ .
- 3) Вызвать вспомогательный алгоритм *transfusion*.
- 4) Конец алгоритма.



## \* Вспомогательный рекурсивный алгоритм *transfusion* (версия 0)

- 1) Начало алгоритма.
- 2) Вывести значения наполненности сосудов  $A$ ,  $B$ ,  $C$ .
- 3) Если  $A = 0$  или  $B = 0$  или  $C = 0$ , то конец алгоритма; иначе перейти к следующему шагу.
- 4) Если  $A \geq B$ , то вывести «-12->»,  $A \leftarrow A - B$ ,  $B \leftarrow 2B$ , вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.
- 5) Если  $A \geq C$ , то вывести «-13->»,  $A \leftarrow A - C$ ,  $C \leftarrow 2C$ , вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.
- 6) Если  $B \geq A$ , то вывести «-21->»,  $B \leftarrow B - A$ ,  $A \leftarrow 2A$ , вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.
- 7) Если  $B \geq C$ , то вывести «-23->»,  $B \leftarrow B - C$ ,  $C \leftarrow 2C$ , вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.
- 8) Если  $C \geq A$ , то вывести «-31->»,  $C \leftarrow C - A$ ,  $A \leftarrow 2A$ , вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.
- 9) Если  $C \geq B$ , то вывести «-32->»,  $C \leftarrow C - B$ ,  $B \leftarrow 2B$ , вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.
- 10) Конец алгоритма.

## \* Процессы переливаний по алгоритму *transfusion* версии 0

**Вариант 1)**  $(8, 3, 11) -12-\rightarrow (5, 6, 11) -21-\rightarrow (10, 1, 11) -12-\rightarrow (9, 2, 11) -12-\rightarrow (7, 4, 11) -12-\rightarrow (3, 8, 11) -21-\rightarrow (6, 5, 11) -12-\rightarrow (1, 10, 11) -21-\rightarrow (2, 9, 11) -21-\rightarrow (4, 7, 11) -21-\rightarrow (8, 3, 11)$  (бесконечный цикл, количество переливаний  $n = \infty$ ).

**Вариант 2)**  $(10, 2, 6) -12-\rightarrow (8, 4, 6) -12-\rightarrow (4, 8, 6) -21-\rightarrow (8, 4, 6)$  (бесконечный цикл, количество переливаний  $n = \infty$ ).

**Вариант 3)**  $(11, 14, 3) -13-\rightarrow (8, 14, 6) -13-\rightarrow (2, 14, 12) -21-\rightarrow (4, 12, 12) -21-\rightarrow (8, 8, 12) -12-\rightarrow (0, 16, 12)$  (успех, количество переливаний  $n = 5$ ).

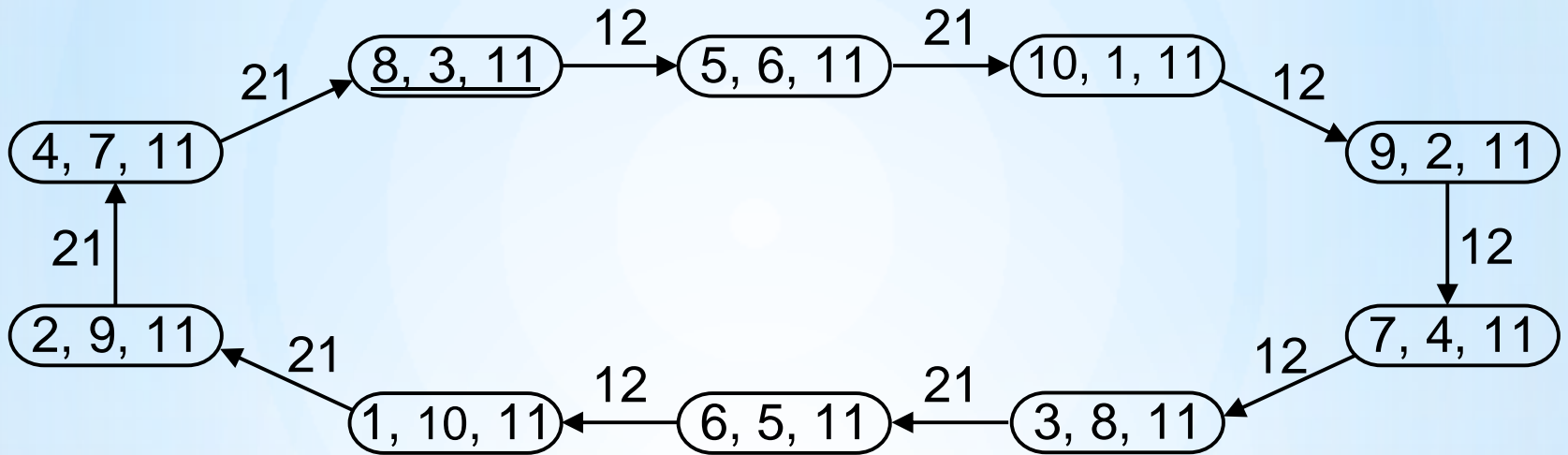
**Вариант 4)**  $(4, 7, 2) -13-\rightarrow (2, 7, 4) -21-\rightarrow (4, 5, 4) -13-\rightarrow (0, 5, 8)$  (успех,  $n = 3$ ).

**Вариант 5)**  $(14, 5, 8) -12-\rightarrow (9, 10, 8) -13-\rightarrow (1, 10, 16) -21-\rightarrow (2, 9, 16) -21-\rightarrow (4, 7, 16) -21-\rightarrow (8, 3, 16) -12-\rightarrow (5, 6, 16) -21-\rightarrow (10, 1, 16) -12-\rightarrow (9, 2, 16) -12-\rightarrow (7, 4, 16) -12-\rightarrow (3, 8, 16) -21-\rightarrow (6, 5, 16) -12-\rightarrow (1, 10, 16)$  (бесконечный цикл,  $n = \infty$ ).

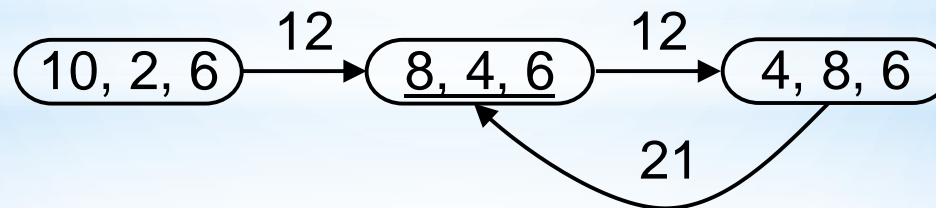
**Вариант 6)**  $(5, 4, 6) -12-\rightarrow (1, 8, 6) -21-\rightarrow (2, 7, 6) -21-\rightarrow (4, 5, 6) -21-\rightarrow (8, 1, 6) -12-\rightarrow (7, 2, 6) -12-\rightarrow (5, 4, 6)$  (бесконечный цикл,  $n = \infty$ ).

**Вариант 7)**  $(7, 6, 12) -12-\rightarrow (1, 12, 12) -21-\rightarrow (2, 11, 12) -21-\rightarrow (4, 9, 12) -21-\rightarrow (8, 5, 12) -12-\rightarrow (3, 10, 12) -21-\rightarrow (6, 7, 12) -21-\rightarrow (12, 1, 12) -12-\rightarrow (11, 2, 12) -12-\rightarrow (9, 4, 12) -12-\rightarrow (5, 8, 12) -21-\rightarrow (10, 3, 12) -12-\rightarrow (7, 6, 12)$  (бесконечный цикл,  $n = \infty$ ).

# \* Взвешенные графы процессов переливаний для вариантов 1 и 2

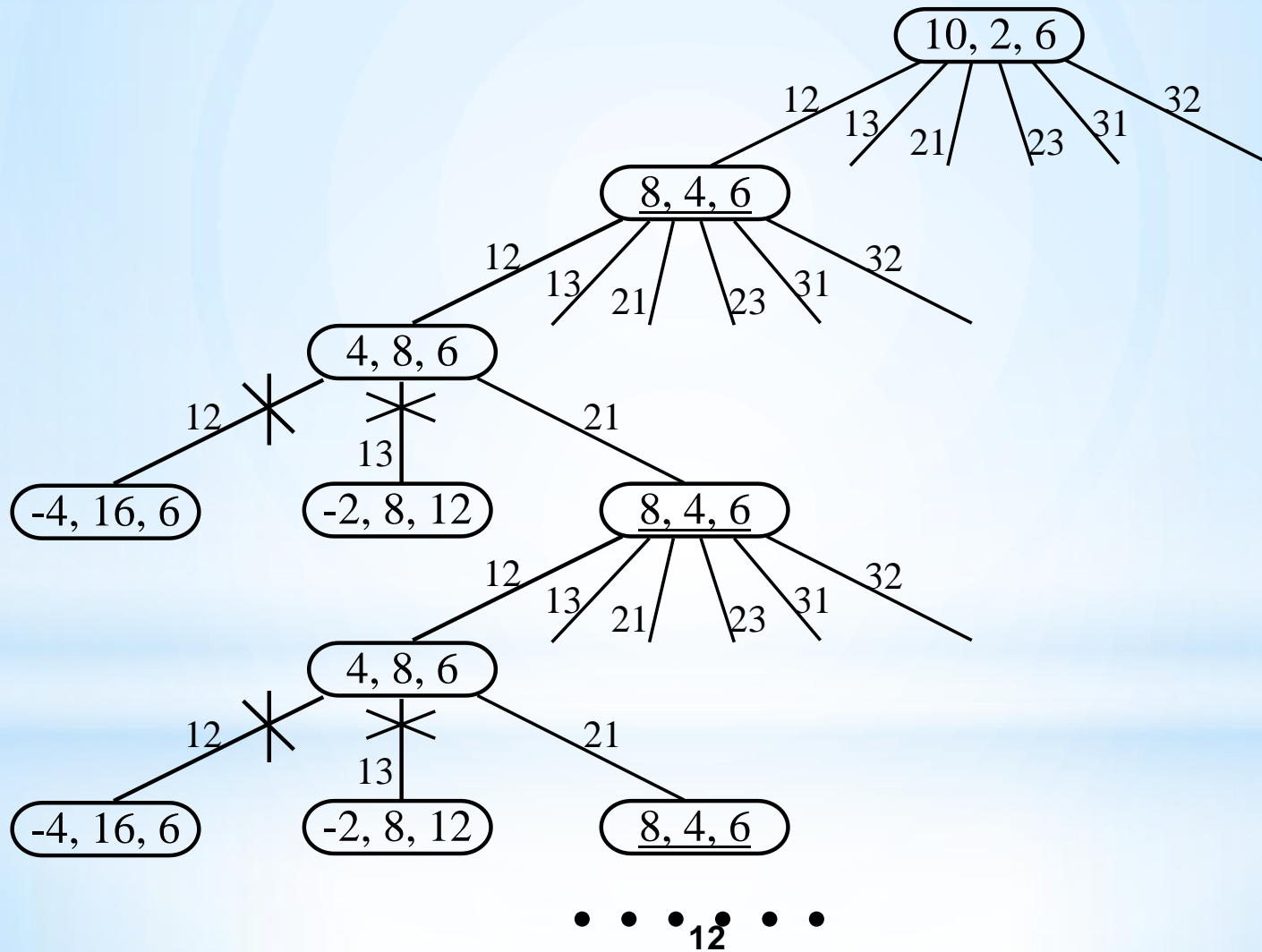


Вариант 1



Вариант 2

\* **Бесконечно высокое дерево рекурсивных вызовов для варианта 2**



# \* Вспомогательный рекурсивный алгоритм *transfusion* (версия 1)

- 1) Начало алгоритма.
- 2) Вывести значения наполненности сосудов  $A$ ,  $B$ ,  $C$ .
- 3) Если  $A = 0$  или  $B = 0$  или  $C = 0$ , то конец алгоритма; иначе перейти к следующему шагу.
- 4) Если  $A = B$ , то вывести «-12->»,  $A \leftarrow A - B$ ,  $B \leftarrow 2B$ , вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.
- 5) Если  $A = C$ , то вывести «-13->»,  $A \leftarrow A - C$ ,  $C \leftarrow 2C$ , вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.
- 6) Если  $B = C$ , то вывести «-23->»,  $B \leftarrow B - C$ ,  $C \leftarrow 2C$ , вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.
- 7) Если  $A > B$ , то вывести «-12->»,  $A \leftarrow A - B$ ,  $B \leftarrow 2B$ , вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.
- .....
- .....
- 12) Если  $C > B$ , то вывести «-32->»,  $C \leftarrow C - B$ ,  $B \leftarrow 2B$ , вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.
- 13) Конец алгоритма.

## \* Процессы переливаний по алгоритму *transfusion* версии 1

**Вариант 1)**  $(\underline{8, 3, 11}) -12-\rightarrow (5, 6, 11) -21-\rightarrow (10, 1, 11) -12-\rightarrow (9, 2, 11) -12-\rightarrow (7, 4, 11) -12-\rightarrow (3, 8, 11) -21-\rightarrow (6, 5, 11) -12-\rightarrow (1, 10, 11) -21-\rightarrow (2, 9, 11) -21-\rightarrow (4, 7, 11) -21-\rightarrow (\underline{8, 3, 11})$  (бесконечный цикл, количество переливаний  $n = \infty$ ).

**Вариант 2)**  $(10, 2, 6) -12-\rightarrow (\underline{8, 4, 6}) -12-\rightarrow (4, 8, 6) -21-\rightarrow (\underline{8, 4, 6})$  (бесконечный цикл, количество переливаний  $n = \infty$ ).

**Вариант 3)**  $(11, 14, 3) -13-\rightarrow (8, 14, 6) -13-\rightarrow (2, 14, 12) -21-\rightarrow (4, 12, 12) -23-\rightarrow (4, 0, 24)$  (успех, количество переливаний  $n = 4$ ).

**Вариант 4)**  $(4, 7, 2) -13-\rightarrow (2, 7, 4) -21-\rightarrow (4, 5, 4) -13-\rightarrow (0, 5, 8)$  (успех,  $n = 3$ ).

**Вариант 5)**  $(14, 5, 8) -12-\rightarrow (9, 10, 8) -13-\rightarrow (\underline{1, 10, 16}) -21-\rightarrow (2, 9, 16) -21-\rightarrow (4, 7, 16) -21-\rightarrow (8, 3, 16) -12-\rightarrow (5, 6, 16) -21-\rightarrow (10, 1, 16) -12-\rightarrow (9, 2, 16) -12-\rightarrow (7, 4, 16) -12-\rightarrow (3, 8, 16) -21-\rightarrow (6, 5, 16) -12-\rightarrow (\underline{1, 10, 16})$  (бесконечный цикл,  $n = \infty$ ).

**Вариант 6)**  $(\underline{5, 4, 6}) -12-\rightarrow (1, 8, 6) -21-\rightarrow (2, 7, 6) -21-\rightarrow (4, 5, 6) -21-\rightarrow (8, 1, 6) -12-\rightarrow (7, 2, 6) -12-\rightarrow (\underline{5, 4, 6})$  (бесконечный цикл,  $n = \infty$ ).

**Вариант 7)**  $(7, 6, 12) -12-\rightarrow (1, 12, 12) -23-\rightarrow (1, 0, 24)$  (успех, количество переливаний  $n = 2$ ).

# \* Вспомогательный рекурсивный алгоритм *transfusion* (версия 2)

1) Начало алгоритма.

2) Вывести значения наполненности сосудов  $A$ ,  $B$ ,  $C$ .

3) Если  $A = 0$  или  $B = 0$  или  $C = 0$ , то конец алгоритма; иначе перейти к следующему шагу.

4) Если  $A = B$ , то вывести «-12->»,  $A \leftarrow A - B$ ,  $B \leftarrow 2B$ , вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.

.....

6) Если  $B = C$ , то вывести «-23->»,  $B \leftarrow B - C$ ,  $C \leftarrow 2C$ , вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.

7) Если  $A > B$ , то определить наполненность сосудов в результате предполагаемого переливания  $A \leftarrow A - B$ ,  $B \leftarrow 2B$ , вызвать алгоритм *search\_insert* и если состояние наполненности  $(A, B, C)$  в дереве отсутствует, то вывести «-12->» и вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.

.....

.....

12) Если  $C > B$ , то определить наполненность сосудов в результате предполагаемого переливания  $C \leftarrow C - B$ ,  $B \leftarrow 2B$ , вызвать алгоритм *search\_insert* и если состояние наполненности  $(A, B, C)$  в дереве отсутствует, то вывести «-32->» и вызвать алгоритм *transfusion*; иначе вернуться назад к шагу 8.

13) Конец алгоритма.

## \* Процессы переливаний по алгоритму *transfusion* версии 2

Вариант 1) (8, 3, 11) -12-> (5, 6, 11) -21-> (10, 1, 11) -12-> (9, 2, 11) -12-> (7, 4, 11) -12-> (3, 8, 11) -21-> (6, 5, 11) -12-> (1, 10, 11) -21-> (2, 9, 11) -21-> (4, 7, 11) -31-> (8, 7, 7) -23-> (8, 0, 14) (успех, количество переливаний  $n = 11$ ).

Вариант 2) (10, 2, 6) -12-> (8, 4, 6) -12-> (4, 8, 6) -23-> (4, 2, 12) -12-> (2, 4, 12) -31-> (4, 4, 10) -12-> (0, 8, 10) (успех, количество переливаний  $n = 6$ ).

Вариант 3) (11, 14, 3) -13-> (8, 14, 6) -13-> (2, 14, 12) -21-> (4, 12, 12) -23-> (4, 0, 24) (успех, количество переливаний  $n = 4$ ).

Вариант 4) (4, 7, 2) -13-> (2, 7, 4) -21-> (4, 5, 4) -13-> (0, 5, 8) (успех,  $n = 3$ ).

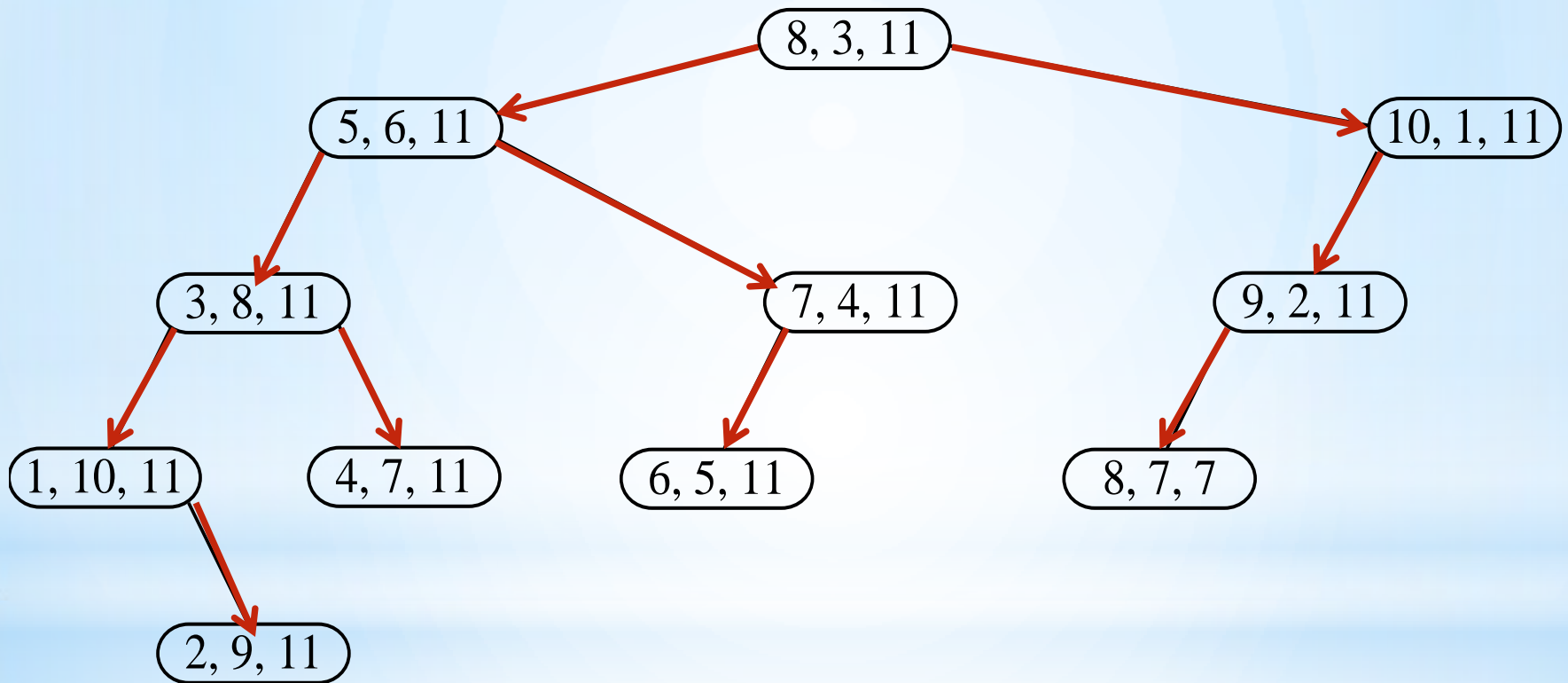
Вариант 5) (14, 5, 8) -12-> (9, 10, 8) -13-> (1, 10, 16) -21-> (2, 9, 16) -21-> (4, 7, 16) -21-> (8, 3, 16) -12-> (5, 6, 16) -21-> (10, 1, 16) -12-> (9, 2, 16) -12-> (7, 4, 16) -12-> (3, 8, 16) -21-> (6, 5, 16) -31-> (12, 5, 10) -12-> (7, 10, 10) -23-> (7, 0, 20) (успех,  $n = 14$ ).

Вариант 6) (5, 4, 6) -12-> (1, 8, 6) -21-> (2, 7, 6) -21-> (4, 5, 6) -21-> (8, 1, 6) -12-> (7, 2, 6) -13-> (1, 2, 12) -21-> (2, 1, 12) -31-> (4, 1, 10) -12-> (3, 2, 10) -12-> (1, 4, 10) -21-> (2, 3, 10) -31-> (4, 3, 8) -12-> (1, 6, 8) -21-> (2, 5, 8) -32-> (2, 10, 3) -21-> (4, 8, 3) -21-> (8, 4, 3) (неудача,  $n = 17$ ).

Вариант 7) (7, 6, 12) -12-> (1, 12, 12) -23-> (1, 0, 24) (успех, количество переливаний  $n = 2$ ).



\* Формируемое по алгоритму *search\_insert* бинарное дерево состояний наполненности сосудов для варианта 1









## \* Вспомогательный рекурсивный алгоритм *transfusion* (версия 3)

- 1) Начало алгоритма.
- 2) Если  $A = 0$  или  $B = 0$  или  $C = 0$ , то конец алгоритма; иначе перейти к следующему шагу.
- 3) Если  $A = B$ , то вывести «-12->»,  $A \leftarrow A - B$ ,  $B \leftarrow 2B$ , вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.
- .....
- .....
- 5) Если  $B = C$ , то вывести «-23->»,  $B \leftarrow B - C$ ,  $C \leftarrow 2C$ , вызвать алгоритм *transfusion*; иначе перейти к следующему шагу.
- 6) Если  $A < B$  и  $B < C$ , то выполнить шаги 7 и 8; иначе перейти к шагу 15.
- 7)  $q \leftarrow B \text{ div } A$ .
- 8) Пока  $q \neq 0$  выполнить шаги 9, 10, ..., 13.
- 9)  $p \leftarrow q \text{ mod } 2$ .
- 10) Если  $p = 1$ , то вывести «-21->»,  $B \leftarrow B - A$ ; иначе вывести «-31->»,  $C \leftarrow C - A$ ,  $A \leftarrow 2A$ .
- 11) Если  $A = C$ , то вывести «-13->»,  $A \leftarrow A - C$ ,  $C \leftarrow 2C$ , вызвать алгоритм *transfusion*,  $q \leftarrow 0$ .
- 12) Если  $B = C$ , то вывести «-23->»,  $B \leftarrow B - C$ ,  $C \leftarrow 2C$ , вызвать алгоритм *transfusion*,  $q \leftarrow 0$ .
- 13)  $q \leftarrow q \text{ div } 2$ .
- 14) Если  $A \neq 0$  и  $B \neq 0$  и  $C \neq 0$ , то вызвать алгоритм *transfusion*.

## \* Окончание вспомогательного рекурсивного алгоритма *transfusion* (версия 3)

- 15) Если  $A < C$  и  $C < B$ , то выполнить шаги 16 и 17; иначе перейти к шагу 24.
  - 16) Переменной  $q$  присвоить результат целочисленного деления  $C$  на  $A$ .
  - 17) Пока  $q \neq 0$  выполнить шаги 18, 19, ..., 22.
  - 18) Переменной  $r$  присвоить значение остатка от целочисленного деления  $q$  на 2.
  - 19) Если  $r = 1$ , то вывести «-31->»,  $C \leftarrow C - A$ ; иначе вывести «-21->»,  $B \leftarrow B - A$ ,  $A \leftarrow 2A$ .
  - 20) Если  $A = B$ , то вывести «-12->»,  $A \leftarrow A - B$ ,  $B \leftarrow 2B$ , вызвать алгоритм *transfusion*,  $q \leftarrow 0$ .
  - 21) Если  $C = B$ , то вывести «-32->»,  $C \leftarrow C - B$ ,  $B \leftarrow 2B$ , вызвать алгоритм *transfusion*,  $q \leftarrow 0$ .
  - 22) Переменной  $q$  присвоить результат целочисленного деления  $q$  на 2.
  - 23) Если  $A \neq 0$  и  $B \neq 0$  и  $C \neq 0$ , то вызвать алгоритм *transfusion*.
- Шаги 24, 25, ..., 32 аналогичны шагам 15, 16, ..., 23.
- Шаги 33, 34, ..., 41 аналогичны шагам 24, 25, ..., 32.
- Шаги 42, 43, ..., 50 аналогичны шагам 33, 34, ..., 41.
- Шаги 51, 52, ..., 59 аналогичны шагам 42, 43, ..., 50.
- 60) Конец алгоритма.

## \* Процессы переливаний по алгоритму *transfusion* версии 3

Вариант 1) (8, 3, 11) -32-> (8, 6, 8) -13-> (0, 6, 16) (успех, количество переливаний  $n = 2$ ).

Вариант 2) (10, 2, 6) -32-> (10, 4, 4) -32-> (10, 8, 0) (успех, количество переливаний  $n = 2$ ).

Вариант 3) (11, 14, 3) -13-> (8, 14, 6) -13-> (2, 14, 12) -21-> (4, 12, 12) -32-> (4, 24, 0) (успех, количество переливаний  $n = 4$ ).

Вариант 4) (4, 7, 2) -23-> (4, 5, 4) -13-> (0, 5, 8) (успех,  $n = 2$ ).

Вариант 5) (14, 5, 8) -32-> (14, 10, 3) -23-> (14, 7, 6) -23-> (14, 1, 12) -12-> (13, 2, 12) -12-> (11, 4, 12) -32-> (11, 8, 8) -32-> (11, 16, 0) (успех,  $n = 7$ ).

Вариант 6) (5, 4, 6) -12-> (1, 8, 6) -21-> (2, 7, 6) -31-> (4, 7, 4) -31-> (8, 7, 0) (успех,  $n = 4$ ).

Вариант 7) (7, 6, 12) -12-> (1, 12, 12) -23-> (1, 0, 24) (успех, количество переливаний  $n = 2$ ).

## \* Серии переливаний для варианта 6

Первая серия - отливание из сосуда 1 средней наполненности 5 мл максимального объёма 4 мл, однократного объёму в сосуде 2 с наименьшей наполненностью 4 мл:

$$(5, 4, 6) [5 \text{ div } 4 = 1 \text{ (нечетное)}] -12 \rightarrow (1, 8, 6).$$

Вторая серия - отливание из сосуда 3 средней наполненности 6 мл максимального объёма 6 мл, шестикратного объёму в сосуде 1 с наименьшей наполненностью 1 мл:

$$(1, 8, 6) [6 \text{ div } 1 = 6 \text{ (четное)}] -21 \rightarrow$$

$$(2, 7, 6) [6 \text{ div } 2 = 3 \text{ (нечетное)}] -31 \rightarrow$$

$$(4, 7, 4) [4 \text{ div } 4 = 1 \text{ (нечетное)}] -31 \rightarrow (8, 7, 0).$$



# \* Характеристики рекурсивного алгоритма *transfusion* версий 0, 1, 2 и 3

№ варианта	Изначальная наполненность сосудов			Результативность решения задачи и число выполненных переливаний для вспомогательного рекурсивного алгоритма			
	A	B	C	версия 0	версия 1	версия 2	версия 3
1	8	3	11	неудача, $n = \infty$	неудача, $n = \infty$	успех, $n = 11$	успех, $n = 2$
2	10	2	6	неудача, $n = \infty$	неудача, $n = \infty$	успех, $n = 6$	успех, $n = 2$
3	11	14	3	успех, $n = 5$	успех, $n = 4$	успех, $n = 4$	успех, $n = 4$
4	4	7	2	успех, $n = 3$	успех, $n = 3$	успех, $n = 3$	успех, $n = 2$
5	14	5	8	неудача, $n = \infty$	неудача, $n = \infty$	успех, $n = 14$	успех, $n = 7$
6	5	4	6	неудача, $n = \infty$	неудача, $n = \infty$	неудача, $n = 17$	успех, $n = 4$
7	7	6	12	неудача, $n = \infty$	успех, $n = 2$	успех, $n = 2$	успех, $n = 2$

## \* Выводы

- 1) Изучена постановка олимпиадной задачи «Переливания».
- 2) Выполнен первичный анализ процесса переливаний с целью освобождения одного из трёх сосудов.
- 3) Спроектированы четыре версии рекурсивного алгоритма решения олимпиадной задачи «Переливания». При этом каждая его более новая версия дает решение олимпиадной задачи не хуже предыдущей.
- 4) Алгоритм версии 3 всегда гарантированно находит кратчайший путь переливаний, который ведёт от изначальной наполненности сосудов к освобождению одного из них, или, по крайней мере, не явно сократимый путь.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. [Олимпиады по программированию](http://www.olympiads.ru/zaoch/2006/problems/l.shtml). I Всероссийская заочная олимпиада школьников по информатике [Электронный ресурс]: Задача L. Переливания. - Режим доступа: <http://www.olympiads.ru/zaoch/2006/problems/l.shtml>.
2. Уайс, М.А. Организация структур данных и решение задач на С++ / М.А. Уайс. - М. : ЭКОМ Паблишерз, 2008. - 896 с.
3. Уинер, Р. Язык Турбо Си / Р. Уинер. - М. : Мир, 1991. - 384 с.
4. Кнут, Д. Искусство программирования для ЭВМ. Т. 3. Сортировка и поиск / Д. Кнут. - М. : Мир, 1978. - 846 с.

**СПАСИБО ЗА ВНИМАНИЕ!**