Алгоритмы и структуры данных-1

SET 1. Задача A1.

Осень 2024. Клычков М. Д.

```cpp
#include <vector>

void selectionSort(std::vector<int> &A) {
  for (size_t i = 0; i < A.size(); ++i) {
    size_t minId = i;

    for (size_t j = i + 1; j < A.size(); ++j) {
      if (A[j] < A[minId]) {
        minId = j;
      }
    }

    std::swap(A[minId], A[i]);
  }
}
```

Пункт 1. В качестве инварианта внутреннего цикла по j можно выбрать условие:

$$P_1 := \{minId = \arg\min_{i \leq k < j} A_k\}$$

Приведем краткое обоснование на словах:

Очевидно, что перед заходом в цикл инвариант выполняется: $\arg\min(\{A_i\}) = i = minId$. В теле цикла находится лишь одна конструкция `if`, условие которой выполняется только при $A_j < A_{minId}$. Если утверждение верно, то $min(\{A_i, \ldots, A_j\}) = A_j$, иначе $min(\{A_i, \ldots, A_j\}) = A_{minId}$. После инкремента j снова получается, что $minId = \arg\min_{i \leq k < j} A_k$, то есть инвариант выполняется.

Такой вот индукцией мы кратко доказали, что P_1 подходит в качестве инварианта внутреннего цикла. На самом деле некоторое неявное подобие доказательства с `INIT`, `MNT` и `TRM`, которое будет описано далее.

Пункт 2. В качестве инварианта внешнего цикла по i можно выбрать условие:
$P_2 :=$ подмассив $\{A_k \mid k \in [0; i)\}$ состоит из i наименьших элементов начального массива, упорядоченных в порядке возрастания.

Краткое обоснование:

В теле внешнего цикла инициализируется переменная $minId$, которая после исполнения внутреннего цикла будет содержать индекс наименьшего элемента на хвосте $A[i \ldots n-1]$. Теперь мы ставим на позицию i найденный минимум, который стоит на позиции $minId$. Так, для каждого i мы находим $(i+1)$-ый минимум массива и ставим его на то место, которое он будет занимать в отсортированном массиве. После инкремента выполняется инвариант: подмассив $\{A_k \mid k \in [0; i)\}$ состоит из i наименьших элементов начального массива, упорядоченных в порядке возрастания.

В конце, когда $i = n$, получаем, что весь массив отсортирован по возрастанию.

Пункт 3. Разберем сначала внутренний цикл, а затем внешний.

Внутренний цикл:

`Инвариант:` $P_1 := \{minId = \arg\min_{i \leq k < j} A_k\}$.

`INIT:` Тривиальный случай: $\arg\min(\{A_i\}) = i = minId$.

`MNT:` В теле цикла находится лишь одна конструкция `if`, условие которой выполняется только при $A_j < A_{minId}$.

Если утверждение верно, то $min(\{A_i, \dots, A_j\}) = A_j$, так как

$$min(\{A_i, \dots, A_{j-1}\}) = A_{minId} \wedge A_j < A_{minId}$$

Теперь $minId = j$, после инкремента j снова получается, что $minId = \arg\min\limits_{i \le k < j} A_k$, то есть инвариант выполняется.

Если утверждение неверно, то $min(\{A_i, \dots, A_j\}) = A_{minId}$, так как

$$min(\{A_i, \dots, A_{j-1}\}) = A_{minId} \wedge A_j >= A_{minId}$$

Остается, что $minId = j$, после инкремента j снова получается, что $minId = \arg\min\limits_{i \le k < j} A_k$, то есть инвариант выполняется.

`TRM`: Причиной окончания цикла является нарушение условия $j < n$. Так как на каждом шаге переменная j увеличивалась на 1, то в конце работы цикла $j = n$. Тогда

$$minId = \arg\min_{i \le k < n} A_k$$

что является индексом $(i+1)$-го минимума всего массива.

Внешний цикл:

`Инвариант:` $P_2 :=$ подмассив $\{A_k \mid k \in [0; i)\}$ состоит из i наименьших элементов начального массива, упорядоченных в порядке возрастания.

`INIT`: Тривиальный случай: массив $\{A_k \mid k \in [0; i)\}$ размера 0 упорядочен.

`MNT`: В теле цикла найдется наименьший элемент на хвосте $A[i \dots n-1]$ (согласно доказательству инварианта внутреннего цикла). Теперь мы ставим на позицию i найденный минимум, который стоит на позиции $minId$. Этот минимум будет являться $(i+1)$ (до инкремента), так как на предыдущих итерациях уже были найдены i меньших значений. Таким образом, после инкремента верно, что начало массива $\{A_k \mid k \in [0; i)\}$ состоит из i наименьших элементов начального массива, упорядоченных в порядке возрастания.

`TRM`: Причиной окончания цикла является нарушение условия $i < n$. Так как на каждом шаге переменная i увеличивалась на 1, то в конце работы цикла $i = n$. Тогда полученный после последней итерации исходный массив состоит из n наименьших элементов (то есть всех, так как общее их количество $= n$), упорядоченных по возрастанию, то есть весь массив отсортирован.