электронное научно-техническое издание
НАУКА и ОБРАЗОВАНИЕ
Эл № ФС 77 - 30569. Государственная регистрация №0421100025. ISSN 1994-0408

Операции над упорядоченными множествами

06, июнь 2011
автор: Овчинников В. А.
УДК 004
УДК 004.3 + 519.6

МГТУ им. Н.Э. Баумана
vaovchinnikov@gmail.com

В ходе реализации алгоритмов над графами неоднократно приходится выполнять над представляющими их множествами операции пересечения, объединения, дополнения и др. Асимптотическая оценка вычислительной сложности «в худшем» этих операций над неупорядоченными множествами X и Y равна $O(n{\cdot}m)$, где $n = |X|$, $m = |Y|$. В [1] показано, что операции пересечения, объединения и дополнения над упорядоченными множествами требуют меньшего количества операций сравнения, что позволит снизить вычислительную сложность алгоритмов. Однако остальные операции там исследованы не были.

Будем считать, что множества заданы перечислением элементов, а элементы представлены их номерами в универсуме. Упорядочивание множеств в данном случае рассматривается как сортировка номеров, т. е. целых чисел, например по их возрастанию. Для сравнения эффективности операций над неупорядоченными и упорядоченными множествами будем оценивать их вычислительную сложность по количеству сравнений элементов множеств. Проверку условий выхода из циклов учитывать не будем.

Рассмотрим особенности реализации операций над упорядоченными множествами. Упорядоченность элементов множеств X и Y позволит нам в общем случае исключать сравнение их индексов, лежащих в определенных диапазонах. Например, если при проверке условия $x_i = y_j$ выяснится, что x_1 не равно и больше y_8, то все следующие элементы множества X нет необходимости сравнивать с y_1, y_2,…, y_8. Напомним, что для реализации операций пересечения, объединения и дополнения над неупорядоченными множествами в худшем случае требуется сравнение каждого элемента со всеми.

Для получения объективной оценки сверху вычислительной сложности операций над упорядоченными множествами необходимо определить, какие операции отношения номеров следует выполнять, и сконструировать такие наборы данных, которые потребуют максимального числа проверок. Вид операций над номерами и их количество зависят от типа операции над множествами и соотношения целых констант, задающих элементы $x_i \in X$ и $x_j \in Y$. Таким образом, конструирование наборов данных заключается в определении соотношения элементов двух массивов.

Отметим важный факт – множество, полученное в результате выполнения любой операции над упорядоченными множествами, является упорядоченным.

Принадлежность элемента упорядоченному множеству. При поиске $y = x_i$ в упорядоченном множестве методом двоичного деления меньшее количество операций сравнения будет, если проверять данное условие (или инверсное $y \neq x_i$), и при его невыполнении (для инверсного при выполнении) проверять условие $y > x_i$ либо $y < x_i$. Количество операций сравнения будет максимальным, если $y \notin X$, и равно:

$$N_{\text{ср. пр}} = 2\ log_2 n,$$

где $n = |X|$.

Алгоритм выполнения этой операции показан на рис.1.

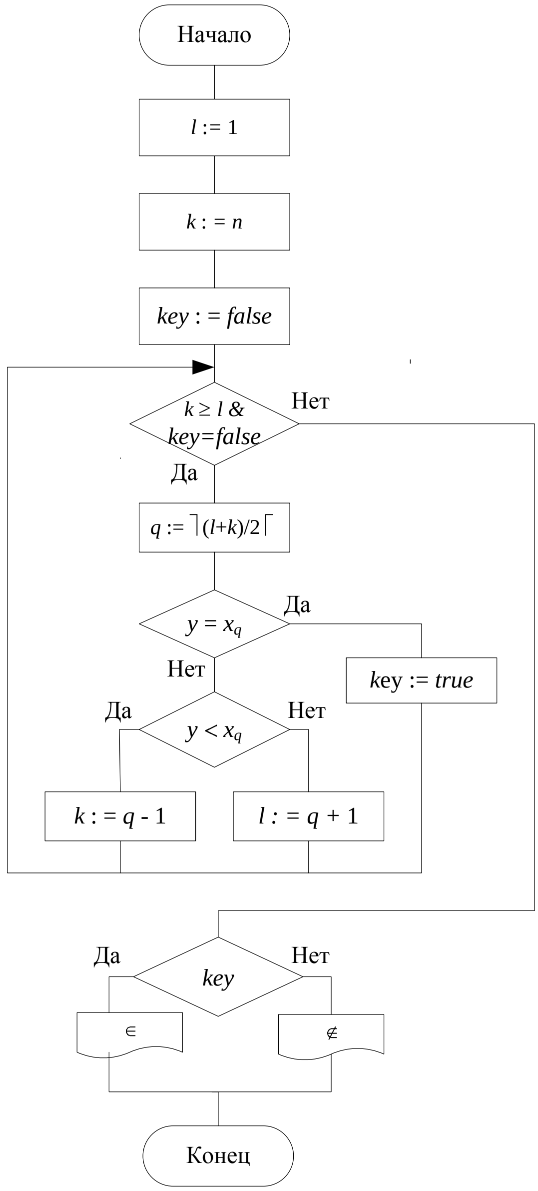

Рис. 1. Схема алгоритма выполнения операции принадлежности элемента упорядоченному множеству

Считая, что сортировка массива, состоящего из n элементов, требует $n\ log_2n$ операций сравнения, получим суммарное количество сравнений, необходимых для предварительного упорядочивания множеств и выполнения операции $y \in X$:

$$N_{\text{ср.}\Sigma} = (n + 2)\ log_2n.$$

Таким образом, асимптотическая оценка вычислительной сложности выполнения операции $y \in X$ над упорядоченным множеством с учётом вычислительной сложности сортировки $O(n\ log_2 n)$. Очевидно, что использование упорядоченного множества при однократном выполнении операции $y \in X$ и отсутствии других операций над этим множеством нецелесообразно. Степень сокращения количества сравнений за счёт упорядочивания, например при n-кратном выполнении этой операции, будет:

$$K = n^2 / 3n\ log_2 n = n / 3\ log_2 n.$$

Реализация операции проверки включения упорядоченных множеств X ⊆ Y. Из определения операции следует, что отношение нестрогого включения не выполняется, если хотя бы для одного элемента $x_i \in X$ не существует y_j такого, что $x_i = y_j$. При сравнении элементов упорядоченных по возрастанию множеств справедливость $x_i \neq y_j$ и $x_i < y_j$ означает, что никакой элемент множества Y не равен x_i. Поскольку возможен вариант, когда $X \setminus x_n \subset Y$ и $x_n = y_m$, условием продолжения сравнения элементов множеств X и Y будет:

$$i \leq n \ \&\ j \leq m \ \&\ r = 0,$$

где $r = 0$ – признак того, что ситуация $x_i \neq y_j$ и $x_i < y_j$ не имела места.

При выполнении условия $x_i = y_j$ необходимо переходить к сравнению следующих элементов рассматриваемых множеств. Алгоритм установления справедливости отношение нестрогого включения множеств X и Y показан на рис. 2.

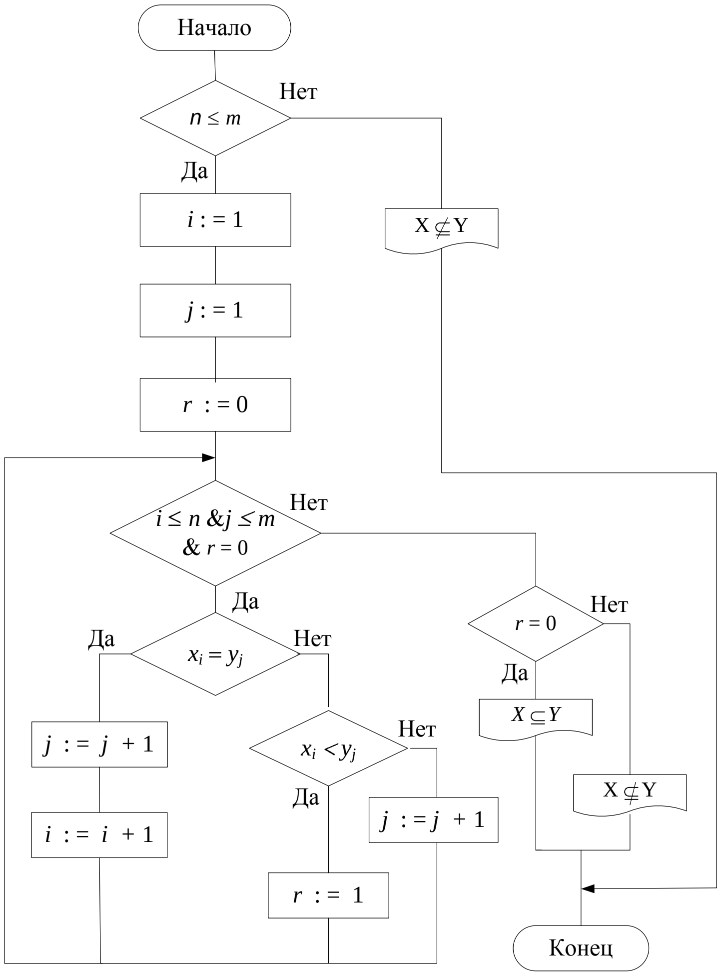

Рис. 2. Схема алгоритма выполнения операции нестрогого включения упорядоченных множеств

Вариантом исходных данных, при котором количество операций сравнения элементов множеств максимально, будет например такой, при котором

$$y_{m-1} < x_1 \,\&\, x_1 \neq y_m.$$

Максимальное количество операций сравнения (оценка в худшем) определяется по формуле

$$N_{\text{ср. вкл}} = 2\,m.$$

Суммарное количество сравнений, требуемых для предварительного упорядочивания множеств и выполнения операции $X \subseteq Y$:

$$N_{\text{ср.}\Sigma} = n{\cdot}log_2 n + m\,(log_2 m + 2).$$

Отсюда, количество операций сравнения, необходимых для выполнения операции $X \subseteq Y$, сократится за счет упорядочивания в

$$K = n\,m \,/\, (n \cdot log_2 n + m\,(log_2 m + 2))$$

раз. При достаточно большом m ($m >> n$)

$$K = n \,/\, log_2 m.$$

Реализация операций отношения строгого включения и равенства упорядоченных множеств $X \subset Y$ и $X = Y$. Алгоритмы выполнения этих операций над упорядоченными множествами аналогичны алгоритму операции $X \subseteq Y$. Очевидно, что вначале необходимо проверять условие $n < m$ и $n = m$ соответственно.

Реализация операции объединения упорядоченных множеств $Z = X \cup Y$. В соответствии с определением операции в ходе сравнения элементов множеств X и Y нам надо записывать в множество Z различные элементы из того и другого множеств, а если элементы совпадают, то записывать один из них. Однако при сопоставлении двух элементов упорядоченных множеств установление факта, что $x_i \neq y_j$, является необходимым, но не достаточным основанием для занесения $x_i \in X$ или $y_j \in Y$ в множество Z. При упорядочивании по возрастанию элемент x_i следует заносить в множество Z, если $x_i < y_j$, а элемент y_j – при $y_j \leq x_i$(или x_i при $x_i \leq y_j$, а y_j – при $y_j < x_i$). Следовательно, кроме отношения равенства нам необходимо проверять и отношение «меньше».

При выполнении условия $x_i > y_j$ следует переходить к сравнению x_i с y_{j+1}. Однако, если для некоторого x_i справедливо $x_i > y_m$, это означает, что все элементы $x_{i+1}, x_{i+1},\ldots, x_n > y_m$. Для обнаружения этой ситуации и выполнения соответствующих операций после выхода из цикла сравнения элементов множеств X и Y необходимо проверять условие $i \leq n$. При выполнении данного условия в множество Z следует записать все оставшиеся элементы множества X, начиная с x_i. Аналогично возможна ситуация, когда $y_j > x_n$. Следовательно, необходимо также проверять условие $j \leq m$ и при его выполнении в множество Z записывать оставшиеся элементы множества Y, начиная с y_j. На рис. 3 показан алгоритм выполнения операции объединения упорядоченных множеств X и Y.

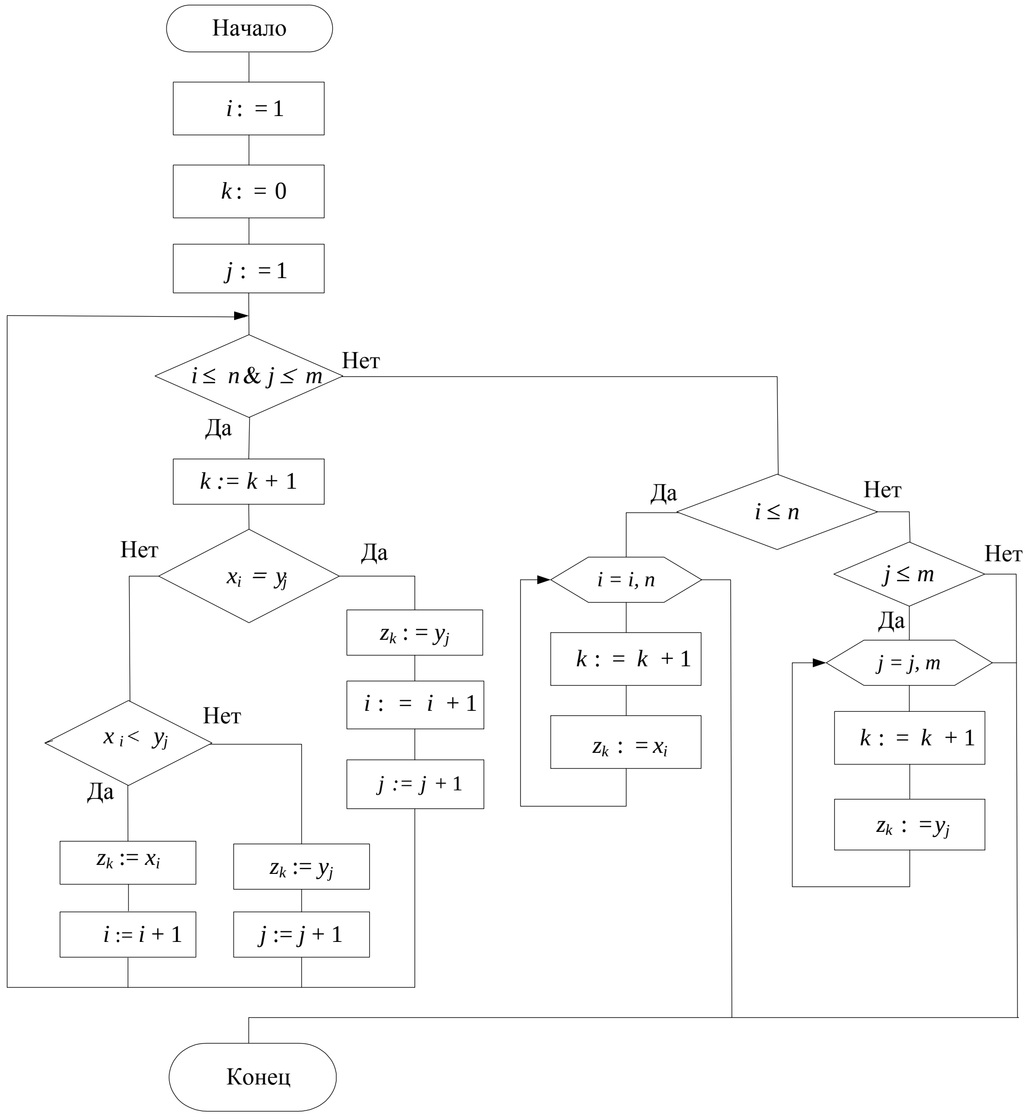

Рис. 3. Схема алгоритма выполнения операции объединения упорядоченных множеств

Обратимся теперь к задаче конструирования худшего варианта данных. При $m > n$ количество операций сравнения будет наибольшим, если 1-й элемент множества X придётся сравнивать со всеми элементами множества Y, а остальные элементы $x_2, x_3,\ldots, x_n$ с элементом y_m. Т.е. худший вариант данных должен удовлетворять условию:

$$(y_{m-1} < x_i < y_m) \;\&\; (x_n \neq y_m),\ i = 1,\ n-1. \qquad (1)$$

Нетрудно видеть, что максимальное количество операций сравнения (оценка в худшем) будет определяться по формуле

$$N_{\text{ср. о}} = 2\,(n + m - 1). \qquad (2)$$

При $|X| = |Y| = n$ суммарное количество сравнений, требуемых для предварительного упорядочивания множеств и выполнения операции объединения:

$$N_{\text{ср.}\Sigma} = 2n \cdot log_2 n + 4n - 2.$$

Отсюда, количество операций сравнения, необходимых для выполнения операции объединения, сократится за счет упорядочивания в

$$K = n^2 / (2n \cdot log_2 n + 4n - 2)$$

раз. При достаточно большом n

$$K = n / log_2 n \qquad (3).$$

Например, $K = 102$ при $n = 1024$ и $K = 1170$ при $n = 16384$.

Реальное сокращение количества операций будет более существенным, чем мы оценили по (3), если операция объединения (и другие, ей подобные) осуществляются в ходе работы алгоритма неоднократно.

Реализация операции пересечения упорядоченных множеств $Z = X \cap Y$. По определению операции в множество Z следует записывать элемент x_i или y_j при выполнении условия $x_i = y_j$. Можно ли при сопоставлении элементов множеств X и ограничится проверкой условия $x_i = y_j$ и переходить к сравнению следующей пары? Нет, этого достаточно только, если $X = Y$, при этом получение Z потребует n операций сравнения, где $n = |X| = |Y|$. Таким образом, $N_{\text{ср.л.}} = n$ – это минимальное количество операций сравнения, т.е. оценка в лучшем. Давайте рассмотрим пример. Пусть $X = <1, 3, 6>$ и $Y = <2, 4, 5>$. Выполнив сравнение $x_i = y_j$ для первых элементов этих массивов, мы определили, что они не равны, следовательно никакой элемент из этой пары пока нельзя заносить в множество Z. Однако мы можем исключить из дальнейшего рассмотрения тот элемент, который имеет меньшее значение (большее, если бы множества были упорядочены по убыванию). Следовательно, кроме операции сравнения $x_i = y_j$ необходимо проверять условие $x_i < y_j$. Таким образом, при каждом сравнении элементов нам в худшем случае придется проверять результаты двух операций $x_i = y_j$ и $x_i < y_j$.

Схема алгоритма, основанного на изложенных соображениях, представлена на рис. 4.

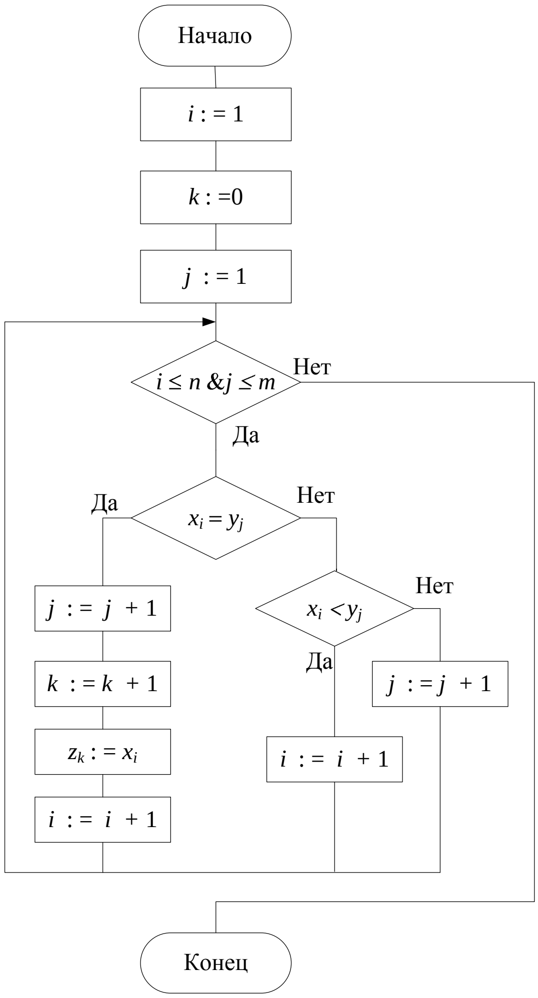

Рис. 4. Схема алгоритма выполнения операции пересечения упорядоченных множеств

Ясно, что количество операций сравнения элементов будет максимальным, если множества X и Y не пересекаются (для общих элементов пересекающихся множеств проверка $x_i < y_j$ не будет выполняться и два элемента – x_i и y_j будут исключены из рассмотрения). Учитывая, что X и Y упорядоченные множества, максимальное количество операций сравнения будет, например при $n < m$, если выполняется условие (1). Нетрудно видеть, что максимальное количество операций сравнения (оценка в худшем) будет определяться по формуле (2).

Реализация операции относительного дополнения множества Y до X*: *Z = X \ Y. По определению операции в множество Z необходимо заносить те элементы множества X, которые не равны никакому элементу множества Y. Следовательно, алгоритм этой операции должен отличаться от алгоритма операции $Z = X \cup Y$ только тем, что в множество Z не следует записывать элементы множества Y (см. рис. 5).

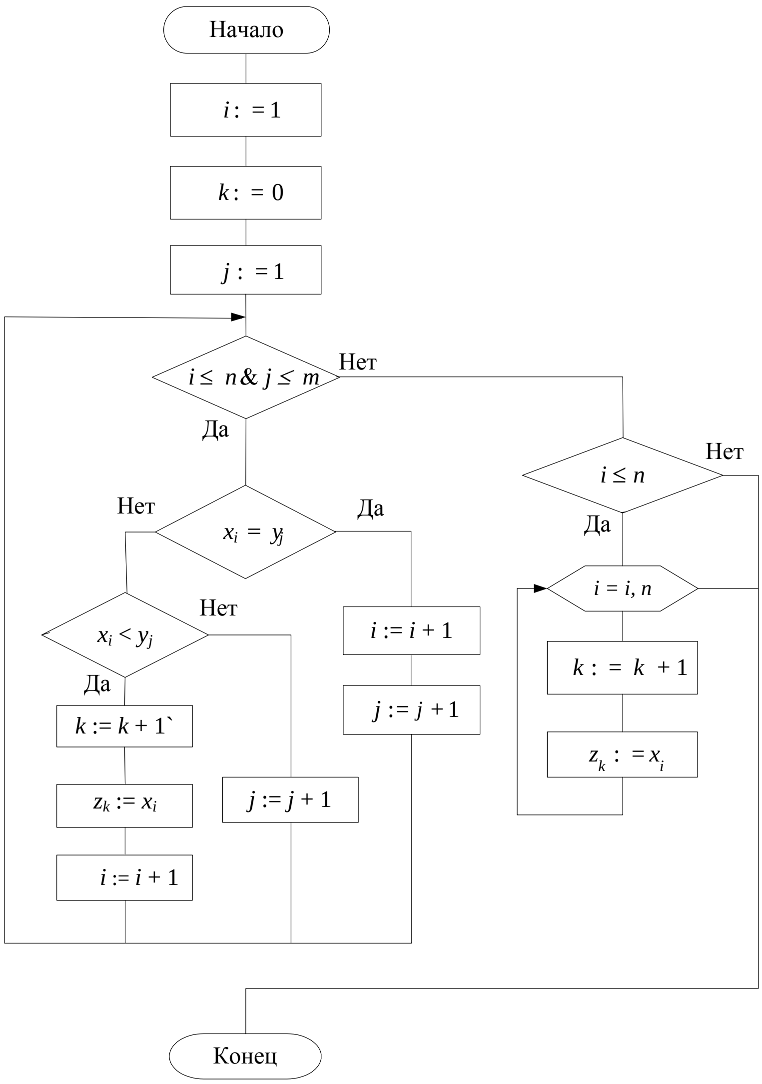

Рис. 5. Алгоритм выполнения операции дополнения упорядоченных множеств

Максимальное количество операций сравнения будет при выполнении условия (1) и оценивается по формуле (2).

Реализация операции симметрической разности упорядоченных множеств X и Y*: *Z = X ∆ Y. В соответствии с определением операции множество Z составляют те элементы $x_i \in X$ и $y_j \in Y$, которые не равны друг другу. Очевидно, что алгоритм выполнения этой операции будет отличаться от алгоритма объединения множеств только тем, что при $x_i = y_j$ не выполняется $z_k := y_j$. Алгоритм реализации операции $Z = X \Delta Y$ представлен на рис. 6.

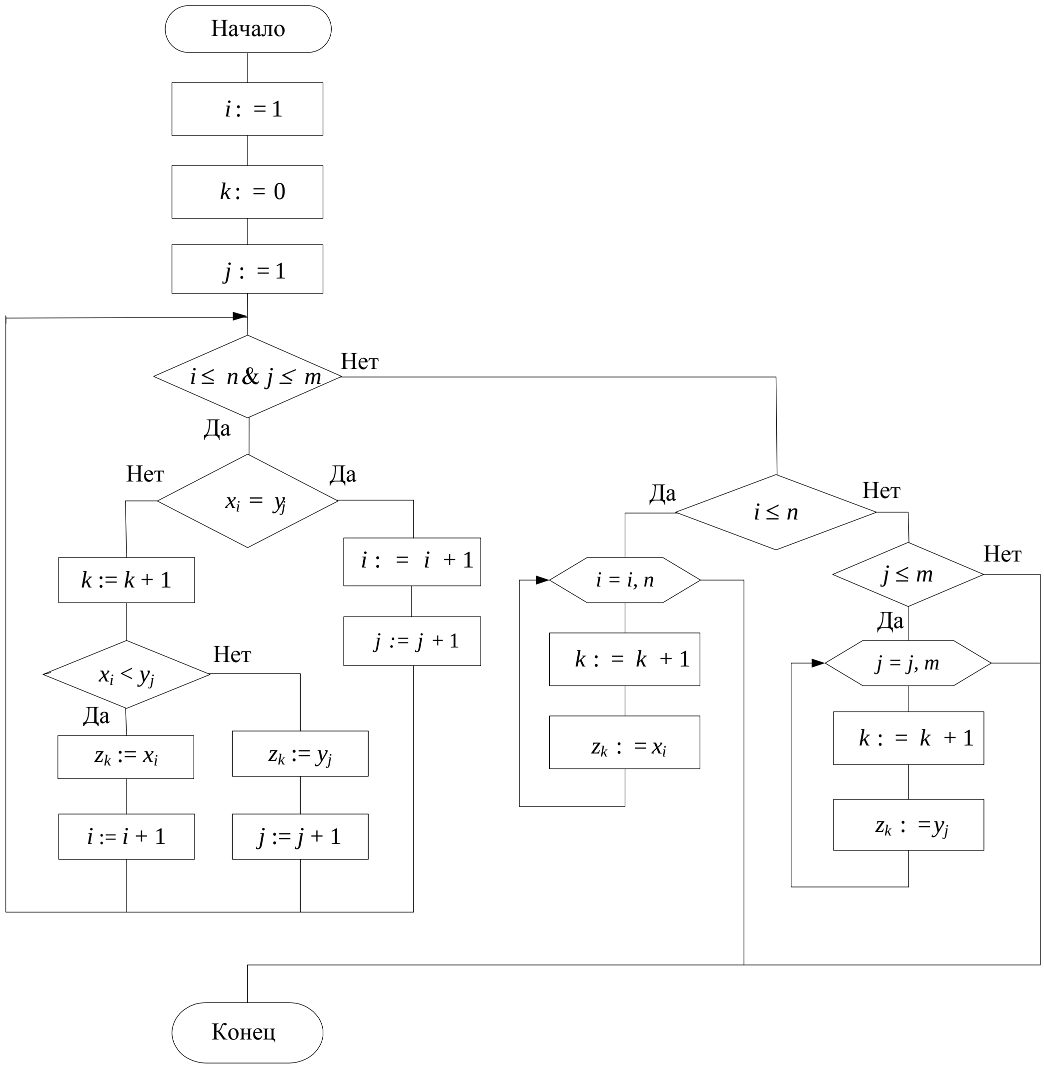

Рис. 6. Алгоритм выполнения операции симметрической разности упорядоченных множеств

Худший вариант исходных данных и вычислительная сложность те же, что и у предыдущей операции.

Литература

1. Овчинников В. А. Алгоритмизация комбинаторно-оптимизационных задач при проектировании ЭВМ и систем: Учеб. для вузов.– М.: Изд-во МГТУ им. Н. Э. Баумана, 2001. – 288с.