Ternary search

Wim H. Hesselink, 2nd July 2003
Dept. of Mathematics and Computing Science, Rijksuniversiteit Groningen
P.O.Box 800, 9700 AV Groningen, The Netherlands
Email: `wim@cs.rug.nl`, Web: `http://www.cs.rug.nl/~wim`

This note is dedicated to the memory of our teacher E.W. Dijkstra, since he would have appreciated the problem and would probably have solved it nicer than we can do below. Our problem is about three ascending sequences of numbers. It was inspired by [2], exercise C-3.1 (p. 213), which is about two ascending sequences.

In order to formulate the problem and its solution, we need some notation on sequences. If s is a sequence of length N, we denote its elements by $s.i$ for $0 \le i < N$. Sequence s is called *ascending* iff $s.(i-1) \le s.i$ for all i with $0 < i < N$. We write $(s|j)$ to denote the initial segment of s with length j, which consists of the elements $s.i$ with $0 \le i < j$. Note that $s.(j-1)$ is the last element of $(s|j)$.

Let a, b, c be three ascending sequences of length N. Let abc be the ordered merge of these sequences, which is an ascending sequence of length $3 \cdot N$. Given is a natural number m. The aim is to construct an efficient algorithm for the determination of numbers i, j, k such that the ordered merge of $(a|i)$, $(b|j)$ and $(c|k)$ equals $(abc|m)$. Note that we allow duplicates or more generally multiple occurrences of elements. This has the effect that the specification of the algorithm is nondeterministic. For simplicity, we assume $m \le N$.

In order to solve the problem, we need to separate the roles of the three sequences as much as possible. We therefore define assertion $A.i$ to mean that the initial segment $(a|i)$ is a subsequence of the initial segment $(abc|m)$. We write $B.j$ and $C.k$ for the analogous assertions on the initial segments $(b|j)$ and $(c|k)$, respectively.

Remark. It is tempting to guess that the conjunction of $A.i$, $B.j$, $C.k$, and $i+j+k=m$ implies that the triple (i,j,k) satisfies the required postcondition. This however is not true. For example, let $a=(0,1,\ldots)$, $b=c=(2,\ldots)$, where the dots stand for numbers ≥ 2. The ordered merge of a, b, and c is of the form $(0,1,2,2,\ldots)$. If we take $m=3$, then $A.1$, $B.1$, and $C.1$ hold, but $(0,2,2)$ is not an initial segment of the ordered merge. □

For simplicity of the invariants, we use the convention that

$$a.(-1) = b.(-1) = c.(-1) = -\infty \ .$$

We claim that

$$(0) \qquad A.i \quad \equiv \quad (\exists\, y,z :: i+y+z \le m \ \wedge \ a.(i-1) \le b.y \ \mathbf{min}\ c.z) \ ,$$

where **min** is the minimum function written as an infix operator, and where the dummies y and z range over the natural numbers.

Formula (0) is proved by a pingpong argument. First, since $A.i$ asserts that $(a|i)$ is a subsequence of $(abc|m)$, it implies the existence of natural numbers y and z such that the ordered merge of $(a|i)$ and $(b|y)$ and $(c|z)$ is a prefix of abc of length $\leq m$. We then have $i+y+z \leq m$ and $a.(i-1) \leq b.y$ and $a.(i-1) \leq c.z$ by the convention for $a.(-1)$ and the definition of the ordered merge.

Conversely, assume the existence of y and z with the properties of (0). Since b and c are ascending, we can choose j and k such that $(b|j)$ and $(c|k)$ are the subsequences of b and c of the elements $< a.(i-1)$. We then have $j \leq y$ and $k \leq z$, and hence $i+j+k \leq m$. Since a is also ascending, the ordered merge of $(a|i)$, $(b|j)$, $(c|k)$ is a prefix of abc. This concludes the proof of (0).

By symmetry, we also have

$$\begin{array}{lcl} B.j & \equiv & (\exists\, x, z :: x+j+z \leq m \ \wedge\ b.(j-1) \leq a.x \ \mathbf{min}\ c.z)\ , \\ C.k & \equiv & (\exists\, x, y :: x+y+k \leq m \ \wedge\ c.(k-1) \leq a.x \ \mathbf{min}\ b.y)\ . \end{array}$$

Lemma 0. Assume that the sequences a, b, c are disjoint: $a.x \neq b.y \neq c.z \neq a.x$ for all x, y, z in $[0\ ..\ N-1]$. Also, assume that $A.i$, $B.j$, $C.k$, and $i+j+k = m$. Then the ordered merge of the initial segments $(a|i)$, $(b|j)$, $(c|k)$ equals $(abc|m)$.

Proof. Let w be the maximum of $a.(i-1)$, $b.(j-1)$, and $c.(k-1)$. It suffices to prove that w is a lower bound of the three numbers $a.i$, $b.j$, $c.k$. By symmetry, we may assume that $w = a[i-1]$. This implies $w \leq a[i]$ since a is ascending. Since $i+j+k = m$, predicate $A.i$ implies the existence of y and z with $y+z \leq j+k$ and $w \leq b.y \ \mathbf{min}\ c.z$. Since w is the maximum of $a.(i-1)$, $b.(j-1)$, and $c.(k-1)$, disjointness of a, b, c implies that $b.(j-1) < w \leq b.y$ and $c.(k-1) < w \leq c.z$. It follows that $j \leq y$ and $k \leq z$. Therefore $y+z \leq j+k$ implies $y = j$ and $z = k$ and hence $w \leq b.j$ and $w \leq c.k$. $\square$

We proceed to derive an algorithm under the assumption that the sequences a, b, c are disjoint. Suppose that $A.i$, $B.j$, and $C.k$ hold with $i+j+k < m$. It seems reasonable to try to increase i, j, or k under preservation of $A.i$, $B.j$, and $C.k$. In order to do this, we can compare elements of the sequences, say $a.p$, $b.q$, $c.r$. In view of the symmetry, we may assume that $a.p \leq b.q$ and $a.p \leq c.r$. If we also have $p+1+q+r \leq m$, this implies $A.(p+1)$, so that i can be replaced by $p+1$.

We introduce program variables i, j, k of type natural with the intended loop invariant

$$J: \quad A.i \ \wedge\ B.j \ \wedge\ C.k \ \wedge\ i+j+k \leq m\ .$$

Lemma 0 says that $J \ \wedge\ i+j+k = m$ implies the required postcondition. This leads us to the program

$$\begin{array}{l} i := j := k := 0\ ; \\ \mathbf{while}\ \ i+j+k \neq m\ \ \mathbf{do}\ S\ \mathbf{end}\ . \end{array}$$

At the repetition, the invariant J is initialized because of the convention on $a.(-1)$, $b.(-1)$, and $c.(-1)$. The required postcondition holds when the algorithm terminates, if ever. In view of the above analysis, we propose the loop body:

$$
\begin{array}{ll}
S: & \textit{choose } p,q,r \textit{ with } p \geq i \ \wedge\ q \geq j \ \wedge\ r \geq k \\
& \quad \wedge\ p+q+r+1 = m \ ; \\
& \textbf{if}\ \ a.p \leq b.q \ \wedge\ a.p \leq c.r \ \ \rightarrow\ \ i := p+1 \ ; \\
& [\!]\ \ b.q \leq a.p \ \wedge\ b.q \leq c.r \ \ \rightarrow\ \ j := q+1 \ ; \\
& [\!]\ \ c.r \leq a.p \ \wedge\ c.r \leq b.q \ \ \rightarrow\ \ k := r+1 \ ; \\
& \textbf{fi}\ \ .
\end{array}
$$

Here we use Dijkstra's guarded commands for the sake of symmetry. Note however that the conditional statement is deterministic because of the assumption that the sequences a, b, c are disjoint.

The choice of p, q, r is possible because of the precondition $i+j+k \leq m-1$ which follows from the guard together with J. Preservation of J follows from the above analysis. Let us define $vf = m - i - j - k$. The repetition terminates since vf remains nonnegative and always decreases. This concludes the proof of correctness.

For the sake of efficiency, we prefer to increase i, j, or k as much as possible. Let $d = (vf-1)$ **div** 3 and $e = (vf-1)$ **mod** 3. Then we have $3{\cdot}d+e+i+j+k = m-1$. So, we can choose $p := i+d$ and $q := j+d$ and $r := k+d+e$. Therefore, vf decreases with at least $d+1 \geq vf/3$. It follows that we have $vf \leq (2/3)^h{\cdot}m$ after h steps of the repetition. Therefore, if $m < (3/2)^h$, the repetition terminates after at most h steps. This proves that the time complexity is logarithmic in m.

We now want to eliminate the assumption that the sequences are disjoint. Given not necessarily disjoint sequences a, b, c, we can make them disjoint by giving the elements $a.i$ a second component 0, the elements $b.j$ a second component 1, the elements $c.k$ a second component 2. In this way, we obtain disjoint sequences a', b', c' of pairs. These sequences are ascending with respect to the lexical order on pairs. Since the lexical order is linear, Lemma 0 is applicable to a', b', c'. Therefore, the algorithm applied to a', b', c' yields numbers i, j, k such that the ordered merge of $(a'|i)$, $(b'|j)$, $(c'|k)$ is a prefix of length m of the ordered merge of a', b', c'.

Let *stripping* be the function that yields the first component of a pair. Since stripping is monotonic for the lexical order on pairs, the stripping of ordered merge of a', b', c' is the ordered merge of a, b, c. Also, the stripping of the ordered merge of $(a'|i)$, $(b'|j)$, $(c'|k)$ is the ordered merge of $(a|i)$, $(b|j)$, $(c|k)$. This shows that the resulting values of i, j, k are correct for the original sequences a, b, c.

We finally eliminate the additional components 0, 1, 2 by observing that

$$
\begin{array}{lllll}
a'.p \leq b'.q & \equiv & (a.p,0) \leq (b.q,1) & \equiv & a.p \leq b.q \ , \\
b'.q \leq a'.p & \equiv & (b.q,1) \leq (a.p,0) & \equiv & b.q < a.p \ ,
\end{array}
$$

and that the analogous rules hold for the combinations (a',c') and (b',c'). It follows that the problem is solved by the algorithm obtained when the guarded command in the loop body S is replaced by

$$
\begin{array}{l}
\textbf{if}\ \ a.p \leq b.q \ \wedge\ a.p \leq c.r \ \ \rightarrow\ \ i := p+1 \ ; \\
[\!]\ \ b.q < a.p \ \wedge\ b.q \leq c.r \ \ \rightarrow\ \ j := q+1 \ ; \\
[\!]\ \ c.r < a.p \ \wedge\ c.r < b.q \ \ \rightarrow\ \ k := r+1 \ ; \\
\textbf{fi}\ \ .
\end{array}
$$

Note that three of the six $\leq$ symbols have been replaced by $<$. The symbols to modify are those where the required order is opposite to the order of the second components (0, 1, 2).

As Jan Eppo Jonker noted, we can finally eliminate the computation of three of the inequalities in the guards by transforming the guarded command into

$$\begin{array}{l}\textbf{if}\ \ a.p \leq b.q \ \wedge\ a.p \leq c.r \ \ \textbf{then}\ \ i := p+1\ ;\\ \textbf{elsif}\ \ b.q \leq c.r \ \ \textbf{then}\ \ j := q+1\ ;\\ \textbf{else}\ \ k := r+1 \ \ \textbf{end}\ .\end{array}$$

Concluding remarks

The main difficulty was to preserve the symmetry between the sequences in order to avoid a combinatorial explosion. On the other hand, a symmetrical solution requires disjointness of the sequences, a requirement we wanted to avoid, and decided to eliminate instead.

We have chosen the name "ternary search" because of the three sequences involved and the analogy of binary search. Just as in Dijkstra's favourite treatment of the binary search [1], the assumption that the sequences are ascending, is not used for preservation of the invariant, but only in the interpretation of the postcondition.

If one wants to weaken the assumption $m \leq N$ to the more natural condition $m \leq 3 \cdot N$, it suffices to extend all three sequences with a sentinel $a.N = b.N = c.N$, which is larger than all values in the sequences. For then the invariant implies that i, j, k are $\leq N$.

The problem and its solution can be easily generalized from three sequences to d sequences with $d \geq 2$. Actually, we found the current version of the proof of Lemma 0 only after formulating this generalization. We don't give the generalization here since it is mainly an exercise in manipulating indices.

References

[1] Dijkstra, E.W.: Constructing the binary search once more. EWD1293. `http://www.cs.utexas.edu/users/EWD/`

[2] Goodrich, M.T., Tamassia, R.: Algorithm Design: foundations, analysis, and internet examples. Wiley, 2002