

20.6.6 Replacement Selection

The last item we will consider is construction of the runs. The strategy used so far is the simplest way: We read as many elements as possible and sort them, writing the result to a tape. This seems like the best approach possible, until we realize that as soon as the first element is written to the output tape, the memory it used becomes available for another element. If the next element on the input tape is larger than the element just output, then it can be included in the run.

Using this observation, we can give an algorithm for producing runs, commonly called *replacement selection*. Initially M elements are read into memory and placed in a priority queue, using `toss` operations, followed by a single `fixHeap`. We perform a `deleteMin`, writing the smallest element to the output tape. We read the next element from the input tape. If it is larger than the element just written, we can add it to the priority queue by an `insert`; otherwise, it cannot go into the current run. Since the priority queue is smaller by one element, this element is stored in the dead space of the priority queue until the run is completed and is used for the next run. Storing an element in the dead space is exactly what was done in `heapsort`. We continue doing this until the size of the priority queue is zero, at which point the run is over. We start a new run by rebuilding a new priority queue with a `fixHeap` operation, in the process using all of the elements in the dead space.

If we are clever, we can make the length of the runs that we initially construct larger than the amount of available main memory. This technique is called *replacement selection*.

	3 Elements in Heap Array			Output	Next Item Read
	array[1]	array[2]	array[3]		
Run 1	11	94	81	11	96
	81	94	96	81	12
	94	96	12	94	35
	96	35	12	96	17
	17	35	12	End of Run	Rebuild
Run 2	12	35	17	12	99
	17	35	99	17	28
	28	99	35	28	58
	35	99	58	35	41
	41	99	58	41	75
	58	99	75	58	15
	75	99	15	75	End of Tape
99		15	99		
Run 3	15			15	

Figure 20.41 Example of run construction