

Foundations of Computer Science

Supervision 4

10. Queues and search strategies

10.1. Conceptual questions

1. Suppose that we have an implementation of queues, based on binary trees, such that each operation takes logarithmic time in the worst case. Outline the advantages and drawbacks of such an implementation compared with one presented in the notes.
2. The traditional way to implement queues uses a fixed-length array. Two indices into the array indicate the start and end of the queue, which wraps around from the end of the array to the start. How appropriate is such a data structure for implementing breadth-first search?
3. Why is iterative deepening inappropriate if $b \approx 1$, where b is the branching factor? What search strategy would make more sense in this case?
4. An interesting variation on the data structures seen in the lectures is a *deque*, or *double-ended queue*¹. A deque supports efficient addition and removal of elements on both ends (either its front or back). Suggest a suitable implementation of deques, justifying your decision.

10.2. Exercises

5. Write a version of the function `breadth` shown on [Page 88](#) using a nested `let` construction rather than `case`.
6. Mathematical sets can be treated as an abstract data type for an unordered collection of unique elements. One approach we may take is to represent a set as an *ordered list* without duplicates: $\{5, 3, 8, 1, 18, 9\}$ would become the OCaml list `[1; 3; 5; 8; 9; 18]`. Code the set operations of membership test, subset test, union and intersection using this ordered-list representation. Remember that you can assume the ordering invariant for the inputs (and thereby make your functions more efficient), and your output should maintain this invariant.

10.3. Optional questions

7. Implement deques as an abstract data type in OCaml (including a type declaration and suitable operations). Estimate the amortised complexity of your solution.

11. Elements of procedural programming

11.1. Conceptual questions

1. Comment, with examples, on the differences between an `int ref list` and an `int list ref`. How would you convert between the two?

¹Deque is usually pronounced "deck", which is convenient because a deck of cards is a good example of a double-ended queue – computer scientists are often way too clever with naming things.

2. What is the effect of `while (C1; B) do C2 done`? Where would such a formulation be useful?

11.2. Exercises

3. Write a version of function `power` (Page 10) using `while` instead of recursion.
4. Write a function to exchange the values of two references, `xr` and `yr`.
5. Arrays of multiple dimensions are represented in OCaml by arrays of arrays. Write functions to (a) create an $n \times n$ identity matrix, given n , and (b) to transpose an $m \times n$ matrix. Identity matrices have the following form:

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$