Outline

Correctness of Recursive Algorithms

Notes
Correctness

- How do we know that our algorithms are correct?

- What does it mean to say an algorithm is “correct”?
Correctness

Some definitions

- Preconditions

- Postconditions
Correctness

Some definitions

- Termination

- Partial correctness
Recursive Binary Search

def recBinSearch(x, A, b, e):
    if b == e:
        if x <= A[b]:
            return b
        else:
            return e + 1
    else:
        m = (b + e) // 2 # midpoint
        if x <= A[m]:
            return recBinSearch(x, A, b, m)
        else:
            return recBinSearch(x, A, m+1, e)
Preconditions and Postconditions for RecBinSearch
Precondition $\Rightarrow$ Termination and Postcondition

Proof: induction on $n = e - b + 1$
Case 1: \( x \leq A[m] \)
Case 2: $x > A[m]$
Precondition ⇒ Termination and Postcondition

Proof: induction on $n = e - b + 1$
Recall MergeSort

MergeSort(A, b, e):
    if b == e: return
    m = (b + e) / 2
    MergeSort(A, b, m)
    MergeSort(A, m+1, e)
    # merge sorted A[b..m] and A[m+1..e] back into A[b..e]
    for i in [b,...,e]: B[i] = A[i]
    c = b
    d = m+1
    for i in [b,...,e]:
        if d > e or (c <= m and B[c] < B[d]):
            A[i] = B[c]
            c = c + 1
        else: # d <= e and (c > m or B[c] >= B[d])
            A[i] = B[d]
            d = d + 1
Preconditions and Postconditions for MergeSort
Prove Precondition $\Rightarrow$ Termination and Postcondition
Prove Precondition $\Rightarrow$ Termination and Postcondition
Algorithm for ClosestPointPairs

ClosestPairRec(P_x, P_y):
    if |P| <= 3:
        find closest points by brute force
    else:
        construct Q_x, Q_y, R_x, R_y
        (q_0,q_1) = ClosestPairRec(Q_x, Q_y)
        (r_0,r_1) = ClosestPairRec(R_x, R_y)
        \( \delta = \min(d(q_0,q_1), d(r_0,r_1)) \)
        (p_0,p_1) = point of (q_0,q_1), (r_0,r_1) with d = \( \delta \)
        L = average of rightmost x-coordinate in Q
        and leftmost x-coordinate in R
        construct S_x, S_y
        for each s ∈ S_y:
            compute distance to next 7 points in S_y
            and let (s_0,s_1) be closest pair found
            if d(p_0,p_1) < d(s_0,s_1): return (p_0,p_1)
            else: return (s_0,s_1)
Preconditions and Postconditions for ClosestPointPairs
Proof?