**Report Section: UT Network**

1. Pseudocode of DP Optimal Run Time

|  |
| --- |
| For t = 0 to n-1:  R[t][1] = ceil ((x[t] – x[0]) /2)  For j = 2 to k:  For t = 0 to n-1:  Best = positive infinity  For I = 0 to t:  Prev = (I > 0) ? R[i-1][j—1] : 0  Block = ceil ( (x[t] - x[i]) / 2)  Best = min(best, cand)  R[t][j] = best  Return R[n-1][k] |

1. Pseudocode Set of Status

|  |
| --- |
| For t = 0 to n-1  R[t][1] = ceil ((c[t] – x[0]) / 2)  CUT[t][1] = 0  For j = 2 to k:  For t = 0 to n-1:  Best = positive infinity  For I = 0 to t:  Prev = ( I > 0) ? R[i-1][j-1] : 0  Block = ceil (( x[t] – x[i]) / 2)  Cand = max (prev, block)  If cand < best or (cand == best && I > best index):  Best = cand  Best Index = i  R[t][j] = best  CUT[t][j] = best index  T = n – 1  J = k  C = empty list  While j >= 1:  i = CUT[t][j]  c = (x[i] + x[t]) /2  Append c to C  T = I - 1  J = j – 1  Reverse C to ascending order  Return C |

1. Runtime and Space

|  |
| --- |
| DP has 3 nested for loops and the split   * Each inner step does o(1) work (ceil and max), so the total time is   o(n^2 \* k)   * Store two n \* k tables, so the space is o(nk)   For example:   * For houses 2,4,6,8 with k = 2 * The optimal response time is 2 achieved by stations 3 and 10 |