Dynamic Programming

Week 10 Shifted Lecture

$B: similar to weight

Ads(n, B)

if (0==n) return 0

// No: Choose do I spend $ I at n^{th}

// Ads(n-1, B) or ?? + Ads(n, B-1)

// Yes: spend $d on n^{th}

$\max_{0 \le d \le B} \{ f_n(d) + Ads(n-1, B-d) \}$ ← O(1)

← O(B) possibilities to examine

O(B) each case

O(nB) cases

Total: $O(nB^2)$

Memoize:

Ads[0...n, 0...B]

for i = 1... n

for j = 0 ... B

Ads[i, j] =

TSP :)

$O(n \cdot n!)$ good if $n \leq 10$

how many cases?

$O(n)$ possibilities

$O(2^n)$ possibilities

} Total: $O(n^2 \cdot 2^n)$

$TSP(i, A)$: shortest path $V_i \rightsquigarrow V_1$ pass through every vertex in A once.

$O(n)$ if memoized previous cases

if $A = \emptyset$

return W_{i1} ← weight edge V_i to V_1

else

// j = ask Shindler, first vertex on journey from i

$$\min_{j \in A} \{ W_{ij} + TSP(j, A - \{j\}) \}$$