# Quiz 8

1) In the lecture you saw the following *subproblem* for solving the *subset sum* problem:

$$T(i,s) = \text{is } s \text{ a subset sum of the array } A[1\ldots i]?$$

True or false: the runtime of the dynamic programming algorithm to solve the subset sum problem that uses this subproblem depends on the size of the entries of $A$?

Select one:

True

False

remember the subset algo, the runtime was $O(n \cdot S)$

2) In the lecture you saw the following *subproblem* for solving the *subset sum* problem:

$$T(i,s) = \text{is } s \text{ a subset sum of the array } A[1\ldots i]?$$

Consider the following dynamic programming table that was made for an array $A[1\ldots3]$ based on this subproblem:

| $T(i,s)$ | $s=0$ | $s=1$ | $s=2$ | $s=3$ | $s=4$ |
|---|---|---|---|---|---|
| $i=0$ | True | False | False | False | False |
| $i=1$ | True | False | True | False | False |
| $i=2$ | True | False | True | True | False |
| $i=3$ | True | False | True | True | True |

Based on this table, which of the following statements are correct? *(Pay attention to the bounds of the arrays!)*

Select one or more:

a. 1 is a subset sum of $A[1\ldots3]$

b. 2 is a subset sum of $A[1\ldots3]$

c. 3 is a subset sum of $A[1\ldots2]$

d. 4 is a subset sum of $A[1\ldots2]$

look at the def of $T(i,s)$
watch out for the indexes

3) In the lecture you saw the following subproblem for solving the *Knapsack* problem:

$$M(i,w) = \text{maximum profit that can be achieved using items in } A[1\ldots i] \text{ of total weight at most } w.$$

Which of the following recursion formulas correctly computes the value of $M(i,w)$?

*(Below, $p_i$ is the profit of item $i$, and $w_i$ is the weight of item $i$.)*

Select one:

a. $M(i,w) = \max\{M(i-1,w),\quad p_i + M(i, w-w_i)\}$

b. $M(i,w) = \max\{M(i-1,w-w_i),\quad p_i + M(i-1, w-w_i)\}$

c. $M(i,w) = \max\{M(i-1,w),\quad p_i + M(i-1, w-w_i)\}$

4) Let $A[1\ldots10]$ be an array of 10 unique integers.

True or false: A longest increasing subsequence in $A[1\ldots5]$ always ends in a strictly smaller number than a longest increasing subsequence in $A[1\ldots10]$.

Select one:

True

False

[1, 2, 3, 4, 999, 5, 6, 7, 8, 9]

A[1...5] : 999

A[1...10] : 9

5) In the lecture you saw the following *subproblem* for solving the *longest increasing subsequence* problem:

$$M(i,\ell) = \text{smallest possible ending of an increasing subsequence of length } \ell \text{ in } A[1\ldots i].$$

Consider the following dynamic programming table that was made for an array $A[1\ldots3]$ based on this subproblem:

| $M(i,\ell)$ | $\ell=1$ | $\ell=2$ | $\ell=3$ |
|---|---|---|---|
| $i=1$ | 4 | $\infty$ | $\infty$ |
| $i=2$ | 4 | 7 | $\infty$ |
| $i=3$ | 4 | ~~8~~ | 8 |

True or false: the table above must contain a mistake.

Select one:

True

False

we could've just end it at 7 (use the previous solution)

# What is used?

## Subset Sum ❗

💡 Idea : Two things can happen to each element

- It gets used in I
- It doesn't get used in I

DP[0...n][0...S]

Definition of the DP table : DP[i][s] = "Can I find a subset sum from A[0...i] that's equal to s"

Computation of an entry :

Initialization : DP[0][0] = True DP[i][0] = True DP[0][s] = False

Recursion :

DP[i][s] = $\underbrace{\text{DP[ i-1 ][ s ]}}_{\text{we don't use i in I}}$ || $\underbrace{\text{DP[ i-1 ][ s-A[i] ]}}_{\text{we use i in I}}$

Extracting the solution : The solution is at DP[n][S]

## Knapsack

💡 Idea : Two things can happen to each element

- We use it and get profit
- We don't use it

DP[0...n][0...W]

Definition of the DP table : DP[i][w] = "Maximum profit from A[0..i] with weight limit w"

Computation of an entry :

Initialization : DP[i][0] = 0

Recursion :

DP[i][w] = $\underbrace{\text{DP[ i-1 ][ w ]}}_{\text{we don't use i in I}}$ || $\underbrace{\text{p[i] + DP[ i-1 ][ w- w[i] ]}}_{\text{we use i in I}}$

Extracting the solution : The solution is at DP[n][W]