1. Suppose you are given an array of numbers, some of which are marked as *icky*, and you want to compute the length of the longest increasing subsequence of $A$ that includes at most $k$ *icky* numbers. Your input consists of the integer $k$, the number array $A[1..n]$, and another boolean array $Icky[1..n]$.

For example, suppose your input consists of the integer $k = 2$ and the following array (with *icky* numbers are indicated by stars):

```
3 ⋆ 1 ⋆ 4 ⋆ 1 ⋆ 5 ⋆ 9 ⋆ 2 ⋆ 6 ⋆ 5 ⋆ 3 ⋆ 5 ⋆ 9 ⋆ 7 ⋆ 9 ⋆ 3 ⋆ 2 ⋆ 3 ⋆ 8 ⋆ 4 ⋆ 6 ⋆ 2 ⋆ 6 ⋆
```

Then your algorithm should return the integer 5, which is the length of the increasing subsequence $4, 5^*, 6, 7, 9^*$.

(a) Describe an algorithm for this problem using dynamic programming.

(b) Describe an algorithm for this problem by reducing it to a standard graph problem.

**Harder problem to think about later:**

2. Let $G$ be a directed acyclic graph whose vertices have labels from some fixed alphabet. Any directed path in $G$ has a label, which is a string obtained by concatenating the labels of its vertices. Recall that a palindrome is a string that is equal to its reversal.

Describe and analyze an algorithm to find the length of the longest palindrome that is the label of a path in $G$. For example, given the dag below, your algorithm should return the integer 6, which is the length of the palindrome **HANNAH**.

```
A
\rightarrow
B
\rightarrow
C
\rightarrow
D
\rightarrow
E
\rightarrow
F
\rightarrow
G
\rightarrow
H
\rightarrow
I
\rightarrow
J
\rightarrow
K
\rightarrow
L
```

(a) Describe an algorithm for this problem using dynamic programming.

(b) Describe an algorithm for this problem by reducing it to a standard graph problem.