2. (a) An example of a problem that the greedy solution does not work for is :

$$I := \{[0,1]w_1 := 1, [0,4]w_2 := 4\}$$

$$p := \{1,2\}$$

The greedy algorithm would pick both of the intervals, when it only needs to pick the second interval. Ratio 1 would be 1 and Ratio 2 would be 4.

(b)

$$\text{int}(i,j,k) = \min \begin{cases} w_i + \text{int}(i+1,x,k) \\ \text{int}(i+1,j,k) \end{cases} \quad x \text{ is the index of the first point not included in } I_i$$

The base cases (couldn't get it to render in Latex in time) would be when the two point indices, j and k, are equal (and we've reached the final point), which would just evaluate to the minimum weight of the intervals that contain this final point. The other would be when we've run out of intervals to choose from $(i = \|I\|)$, which would return 0.

The algorithm is based on the fact that, given a sorted list of points and intervals, iterating through the intervals, at each step we have the choice whether or not to include or exclude the interval. We just pick the min decision.

The function is called by sorting the intervals and the points in increasing order, and calling with $\text{int}(0,0,\|p\|)$.

The memoization structure is just a 3D array, of size $\|I\| \times \|p\| \times \|p\|$, where the evaluation order would be just iterating through the intervals one by one, and iterating through the points at each step.

The overall time complexity is $O(\|I\|\|p\|^2)$.