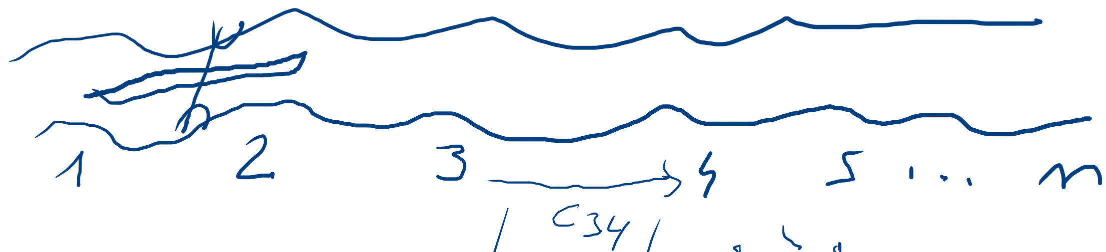
Exercice 3 – Location de canoës



coût c_{ij} pour se rendre de i à j
On recherche une séquence optimale de location pour aller de 1 à n.

Le but de cet exercice est de concevoir un algorithme de programmation dynamique pour résoudre ce problème. On note $opt(i)$ le coût minimum d'une séquence <u>de i à n</u> (avec $1 \leq i \leq n$).

Q 3.1 Donner la valeur du cas de base $opt(n)$, ainsi que la formule de récurrence permettant de déterminer $opt(i)$ (pour $i \in \{1, \ldots, n-1\}$).

$$opt(n) = 0$$

$$opt(i) = \min\{c_{ij} + opt(j) : i < j \leq n\}$$

Q 3.2 On suppose les coûts c_{ij} stockés en machine sous la forme d'une matrice c de taille $n \times n$ ($c_{ij} = c[i][j]$). Ecrire un algorithme de programmation dynamique permettant de déterminer le coût minimal de location de canoë pour aller du site 1 au site n. Cet algorithme pourra utiliser un tableau *opt* de taille n, où la valeur $opt(i)$ sera stockée dans la cellule $opt[i]$.
Indiquer la complexité de votre algorithme.

```
opt[n] ← 0        [ s[n] ← n ]
pour i allant de n-1 à 1 faire
    opt[i] ← +∞
    pour j allant de i+1 à n faire
        si opt[j] + c[i][j] < opt[i] alors
            opt[i] ← opt[j] + c[i][j]        [ s[i] ← j ]
retourner opt[1]
```

Complexité

nb de comparaisons $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 1 = \frac{n(n-1)}{2} \in \Theta(n^2)$

Complexité d'un algo de prog dyn

nb de sous-pb × complexité par sous-pb

$\to n$

$\to O(n) \to n \times O(n) \to O(n^2)$

Q 3.3 Soit $s(i)$ l'arrêt suivant i dans une séquence de coût minimum de i à n. Les valeurs $s(i)$ seront stockées sous la forme d'un tableau $s[1 \ldots n]$. Modifier l'algorithme afin que soient calculées également ces valeurs $s(i)$.

Voir instructions encadrées ci-dessus.

L'objet des questions 4 et 5 est d'appliquer l'algorithme à l'instance numérique de la table 1.

Q 3.4 Appliquer l'algorithme pour déterminer $opt(i)$ et $s(i)$ pour $i \in \{1, \ldots, n\}$.

Q 3.5 En déduire une séquence de coût minimum pour aller de 1 à 8.

coût d'une séquence optimale de 1 à 8

i	1	2	3	4	5	6	7	8
$opt(i)$	9	5	6	3	6	4	1	0
$s(i)$	2	4	4	7	8	7	8	8

Séquence optimale : (1, 2, 4, 7, 8) de coût 9.

	2	**3**	**4**	**5**	**6**	**7**	**8**
1	4	4	7	5	8	11	10
	2	4	2	3	2	5	9
		3	3	1	4	8	12
			4	4	3	2	11
				5	4	8	6
					6	3	7
						7	1

$4 + 2 + 2 + 1 = 9$