

C: TP 11 Programmation en C

DAKKAR Borhen-eddine

Lycée le Corbusier

BTS SN

March 3, 2021

1 Objectifs du TP

Dans ce TP nous allons étudier les algorithmes de tri.

2 Logiciels à utiliser

Sous Linux les outils nécessaire à la compilation du C sont déjà présents, il s'agit de "gcc" et d'un éditeur de texte "gedit" par exemple.

3 Algorithmes de tri

3.1 Tri par sélection

Le tri par sélection est un algorithme de tri par comparaison. Le principe de cet algorithme est le suivant:

- Rechercher le plus petit élément du tableau, et l'échanger avec l'élément d'indice 0;
- Rechercher le second plus petit élément du tableau, et l'échanger avec l'élément d'indice 1;
- Continuer jusqu'à ce que le tableau soit entièrement trié.

Algorithm 1: Tri par sélection

Début

Entier : Tab T [N]

$n \leftarrow \text{Taille}(T)$

$i = 0$

Pour i de 0 à $n-1$ **faire**

$\text{min} = i$

Pour j de $i+1$ à n **faire**

Si $T[j] < T[\text{min}]$ **alors**

$\text{min} = j$

Fin si

Fin pour

Si $\text{min} \neq i$, **alors**

 Echanger $T[i]$ et $T[\text{min}]$

Fin si

Fin pour

FIN

1. Exécutez manuellement ce pseudocode pour le tableau d'entiers suivant:
 $Tab[4] = \{5, 2, 1, 3\}$
2. Ecrivez le pseudocode en langage C.

3.2 Exercice 1 :

Considérons le tableau d'entier suivant:

$$T = \{19, 3, 5, 6, 12, 5, 7, 2, 1\}$$

1. Triez le tableau T par ordre croissant en utilisant le tri par sélection? faites un affichage du tableau avant et après le tri?
2. Changer votre programme maintenant pour qu'il fasse un tri par ordre décroissant.

3.3 Tri par insertion

L'algorithme du tri par **insertion** est un algorithme qui insère un élément dans une liste d'éléments déjà triés. Le jeu de cartes représente un bon exemple de cet algorithme. Dans la main gauche, nous avons des cartes triées de la plus petite à la plus grande, et une carte dans la main droite. Nous cherchons à placer cette carte dans la main gauche de façon à ce qu'elle reste triée ? Il faut donc la placer après les cartes plus petites, et avant les cartes plus grandes. Par exemple, si la main gauche est (1 2 4 5), et que nous avons la carte 3 dans la main droite, nous devons la placer après (1 2) et avant (4 5). En faisant cette opération, nous nous retrouvons avec une main gauche (1 2 3 4 5), qui est encore triée. Pour trier entièrement un ensemble de cartes dans le désordre, il suffit alors de placer toutes ses cartes dans la main droite (la main gauche est donc vide), et d'insérer les cartes une à une dans la main gauche, en suivant la procédure ci-dessus.

Le pseudocode de cet algorithme est donné comme suit:

Algorithm 2: Tri par insertion

Début

Entier : Tab T [N], n ,

$n = \text{Taille}(T)$

Pour i de 0 à n **faire**

 %- - mémoriser $T[i]$ dans x - %

$x = T[i]$

 %- - décaler vers la droite les éléments de $T[0]..T[i-1]$ qui sont

 %- - plus grands que x en partant de $T[i-1]$

$j = i$

Tant que $j > 0$ et $T[j - 1] < x$ **faire**

$T[j] = T[j - 1]$

$j = j - 1$

Fin Tq

 %- - placer x dans le "trou" laissé par le décalage

$T[j] \leftarrow x$

Fin pour

FIN

1. Exécutez manuellement le pseudocode ci-dessus pour le tableau d'entiers suivant: $Tab[4] = \{5, 2, 1, 3, 4\}$
2. Ecrivez le pseudocode en langage C.

3.4 Exercice 2:

Considérons le tableau d'entier suivant:

$T = \{19, 3, 5, 6, 12, 5, 7, 2, 1, 8\}$

1. Triez le tableau T par ordre décroissant en utilisant le tri par insertion? faites un affichage du tableau avant et après le tri?
2. Changez votre programme maintenant pour qu'il fasse un tri par ordre croissant.

3.5 Exercice 3:

1. Ecrivez une fonction **Tri_selection** qui prend comme paramètre un tableau d'entier T et sa taille N . La fonction doit renvoyer un tableau trié en utilisant le tri par sélection. Le prototype de la fonction est le suivant:

```
int *Tri_selection(int T[], int N);
```

Testez votre fonction avec le tableau suivant: $T1 = \{22, 15, 7, 4, 8, 3, 6, 2, 1\}$. Affichez le tableau trié.

2. Nous souhaitons modifier la fonction **Tri_selection** de tel façon qu'elle nous permet de choisir le tri à effectuer, ascendant ou descendant. Modifiez cette fonction et testez la avec le tableau de la question 1.

3.6 Exercice 4:

L'American Standard Code for Information Interchange (Code américain normalisé pour l'échange d'information), plus connu sous l'acronyme ASCII ([aski]), est une norme informatique de codage de caractères apparue dans les années 1960. C'est la norme de codage de caractères la plus influente à ce jour. ASCII définit 128 codes à 7 bits, comprenant 95 caractères imprimables : les chiffres arabes de 0 à 9, les 26 lettres de l'alphabet latin en minuscules et en capitales, et des symboles mathématiques et de ponctuation. La figure 1

1. En se basant sur le tableau du code ASCII, écrivez une fonction **Tri_selection** qui permet de trier le tableau suivant:
 $T = \{E, Z, R, M, N, B, F, A, C, N, Q\}$
2. Triez ensuite le tableau suivant: $T1 = \{e, z, r, m, n, b, f, a, c, n, q\}$

3.7 Exercice 5:

Refaites les questions de l'exercice 4 en utilisant le tri par insertion.

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Figure 1: Tableau code ASCII