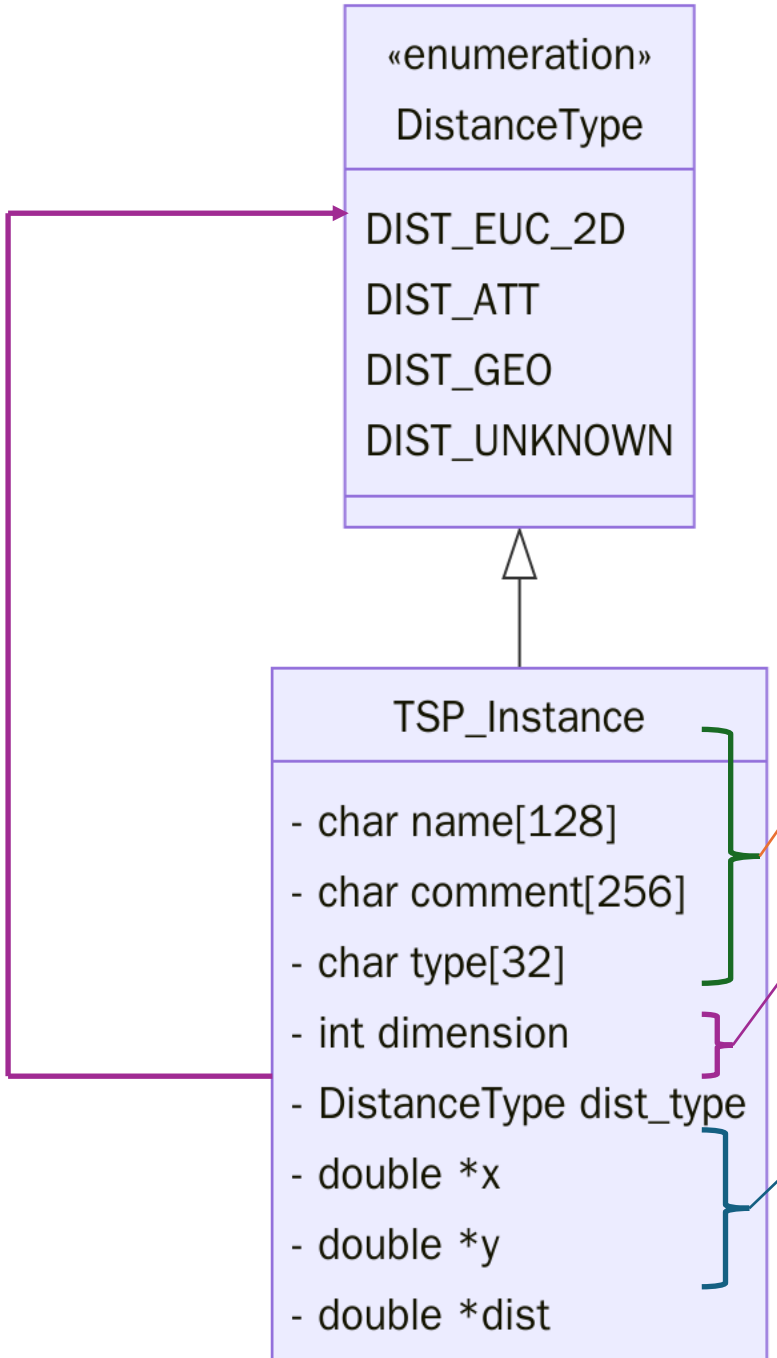
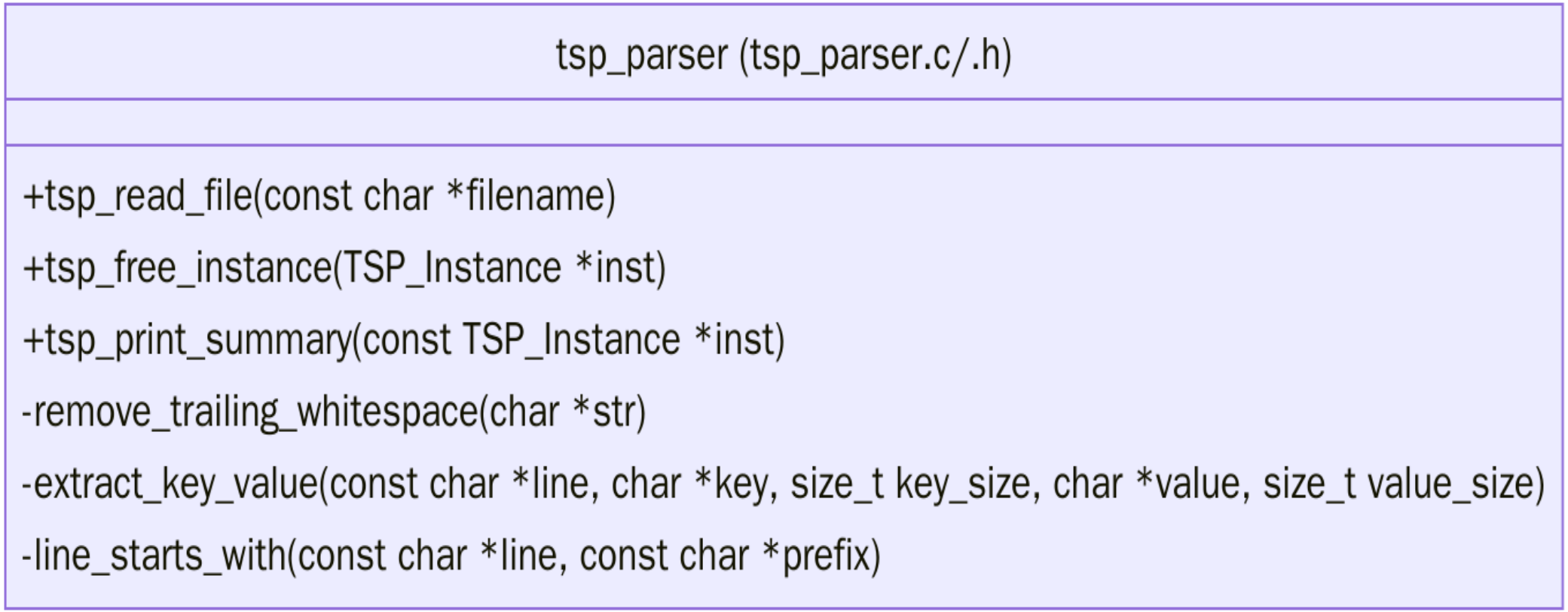

Projet TSP - Phase CC2 🎯
Modules principaux
distance : calculs de distances EUC_2D / ATT / GEO
csv_export : export des résultats en CSV
algo_bf : algorithme brute force exact
main : point d'entrée et orchestration
tsp_types : définitions et structures de base
tsp_parser : lecture et parsing des fichiers TSPLIB
algo_nn : heuristique du plus proche voisin
Architecture du Code 🧩
Structure modulaire .c / .h
Algorithmes Implémentés ⚙️
Brute Force (BF) → exact, petites instances
Nearest Neighbor (NN) → heuristique rapide
Tests CC2 🧪
Comparaison des résultats NN vs BF
CSV générés automatiquement
Cas att10.tsp et att15.tsp validés
Résultats Observés 📊
Tournées valides et cohérentes
Durées d'exécution mesurées
BF plus précis mais plus lent, NN plus rapide
Fonctionnalités CC2 💻
Calcul des distances (DP002)
Algorithmes BF et NN fonctionnels (DP003, DP005)
Export CSV des résultats (DP012)
Parsing complet TSPLIB (DP001)
Interface CLI -f -m -o (DP011)

TSP_Types




```
NAME : att10
COMMENT : 10 premiers de att48)+0
TYPE : TSP
DIMENSION : 10
EDGE_WEIGHT_TYPE : ATT
NODE_COORD_SECTION
1 6734 1453
2 2233 10
3 5530 1424
4 401 841
5 3082 1644
6 7608 4458
7 7573 3716
8 7265 1268
9 6898 1885
10 1112 2049
EOF
```

Tsp_parser


tsp_parser (tsp_parser.c/.h)
+tsp_read_file(const char *filename)
+tsp_free_instance(TSP_Instance *inst)
+tsp_print_summary(const TSP_Instance *inst)
-remove_trailing_whitespace(char *str)
-extract_key_value(const char *line, char *key, size_t key_size, char *value, size_t value_size)
-line_starts_with(const char *line, const char *prefix)

Distance

distance (distance.c/.h)

+parse_distance_type(const char *s)

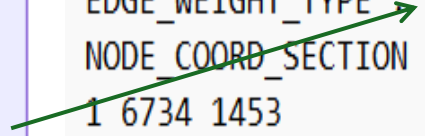
+build_distance_matrix(TSP_Instance *inst)

-dist_euc2d_ij(double xi, double yi, double xj, double yj)

-dist_att_ij(double xi, double yi, double xj, double yj)

-dist_geo_ij(double lati_degmin, double loni_degmin, double latj_degmin, double lonj_degmin)

-geo_tsplib_to_radians(double coord_deg_min, double *out_rad)

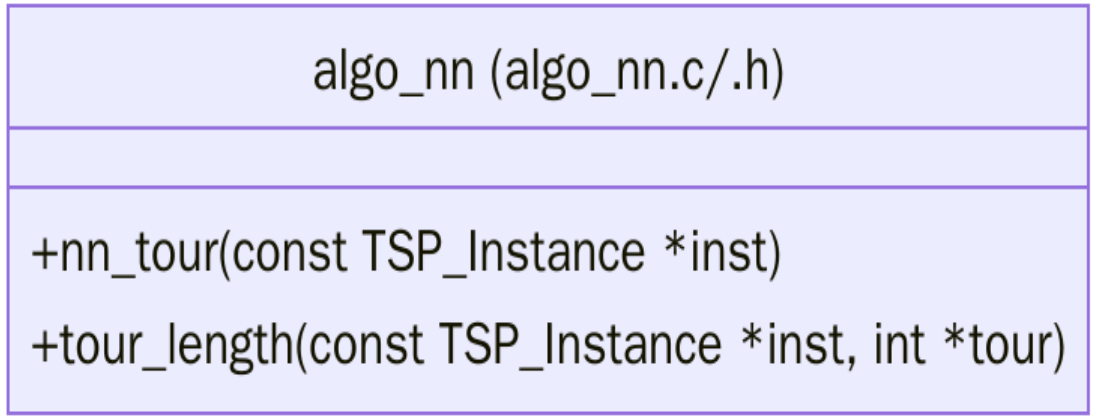


```
NAME : att10
COMMENT : 10 premiers de att48)+0
TYPE : TSP
DIMENSION : 10
EDGE_WEIGHT_TYPE : ATT
NODE_COORD_SECTION
1 6734 1453
2 2233 10
3 5530 1424
4 401 841
5 3082 1644
6 7608 4458
7 7573 3716
8 7265 1268
9 6898 1885
10 1112 2049
EOF
```

Algorithme force brute

algo_bf (algo_bf.c/.h)
+bf_solve(const TSP_Instance *inst, int *best_tour, double *best_cost) -explore(const TSP_Instance *inst, int *current, bool *visited, int pos, int *best_tour, double *best_cost)

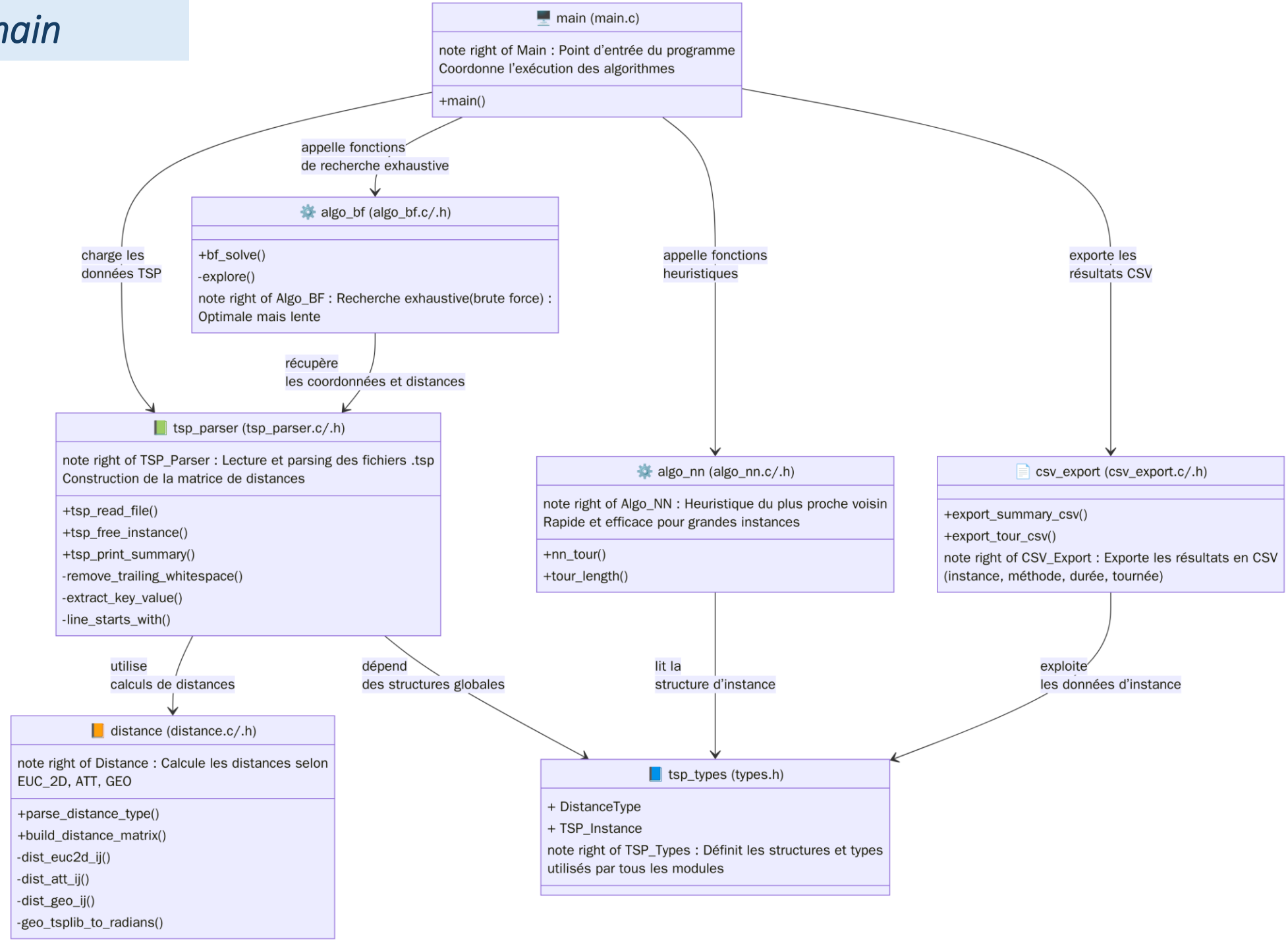
Algorithme plus proche voisin



Export fichier csv

csv_export (csv_export.c/.h)
+export_summary_csv(const char *filename, const char *instance_name, const char *method, double duration_sec, double cost, const int *tour, int n) +export_tour_csv(const char *filename, const int *tour, int n)

Méthode main



Comparaison bf et nn

Force brute (att10.tsp)

```
PS E:\Teamprojekt\tsp_projet_2025> ./tsp.exe -f tests/data/att10.tsp -m bf
M├®thode : bf
Tourn├®e : 1 3 2 4 10 5 6 7 9 8 -1266810639
Longueur : 6178
Dur├®e : 0.036s
```

Plus proche voisin (att15.tsp)

```
PS E:\Teamprojekt\tsp_projet_2025> ./tsp.exe -f tests/data/att15.tsp -m nn
M├®thode : nn
Tourn├®e : 1 9 8 15 12 11 13 14 3 5 2 4 10 7 6 1
Longueur : 7798
Dur├®e : 0.000s
```

Comparaison des performances – Brute Force vs Nearest Neighbor

Critère	Brute Force (att10.tsp)	Nearest Neighbor (att15.tsp)
Méthode utilisée	bf	nn
Nombre de villes	10	15
Tournée obtenue	1 → 3 → 2 → 4 → 10 → 5 → 6 → 7 → 9 → 8 → 1	1 → 9 → 8 → 15 → 12 → 11 → 13 → 14 → 3 → 5 → 2 → 4 → 10 → 7 → 6 → 1
Longueur totale	**6178**	**7798**
Durée d'exécution	**0.036 s**	**0.000 s** (≈ instantané)
Type d'approche	Exhaustive (exacte)	Heuristique (rapide)
Précision	Solution optimale	Solution approchée
Complexité algorithmique	Factorielle (très lente, ~O(n!))	Quadratique (~$O(n^2)$)

Interface CLI

Option	Signification	Exemple d'utilisation
`-f`	Fichier d'entrée au format TSPLIB	`-f tests/data/att10.tsp`
`-m`	Méthode à utiliser (`bf` ou `nn`)	`-m bf`
`-o`	(Optionnel) Fichier CSV d'export des résultats	`-o result.csv`

Exemple:

./tsp.exe -f tests/data/att15.tsp -m nn -o result_nn.csv

Lecture du fichier .tsp → exécution de l'algo NN → export du résultat en CSV

Test export csv

```
PS E:\Teamprojekt\tsp_projet_2025> ./tsp.exe -f tests/data/att15.tsp -m nn -o result_nn.csv
>>
M├®thode : nn
Tourn├®e : 1 9 8 15 12 11 13 14 3 5 2 4 10 7 6 1
Longueur : 7798
Dur├®e : 0.000s
Export├® vers result_nn.csv
```

result_nn.csv

```
1   instance;méthode;durée(s);longueur;tournee
2   att15;nn;0.00;7798;[1,9,8,15,12,11,13,14,3,5,2,4,10,7,6,1]
3
```