Game

- You will create a game with (customized) chess pieces.
- Players start with a predetermined amount of gold.
- There will be many options (different hero) for each piece type.
- Spend your gold on pieces. Higher level *does not mean* higher value – select based on Gold and Attack.
- Maximize total attack points of your pieces.

Max level	Pawn	Rook	Archer	Knight	Bishop	War ship	Siege	Queen	King
1	✓	✗	✗	✗	✗	✗	✗	✗	✗
2	✓	✓	✗	✗	✗	✗	✗	✗	✗
3	✓	✓	✓	✗	✗	✗	✗	✗	✗
...	...	...	...	...	...	...	...	...	...
9	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: *Name of the pieces and their levels (piece types).*

- Players start with a predetermined amount of gold.
- Spend your gold on pieces by selecting the set of hero.
- Maximize total attack points of your pieces.
- You don't have to fill all levels. You cannot select two hero at the same level.

Ex: suppose max level is set to 3, available piece per type is 5.

For *Archer*, you may select one of them:

John

80

500

Robert

90

450

James

75

800

William

50

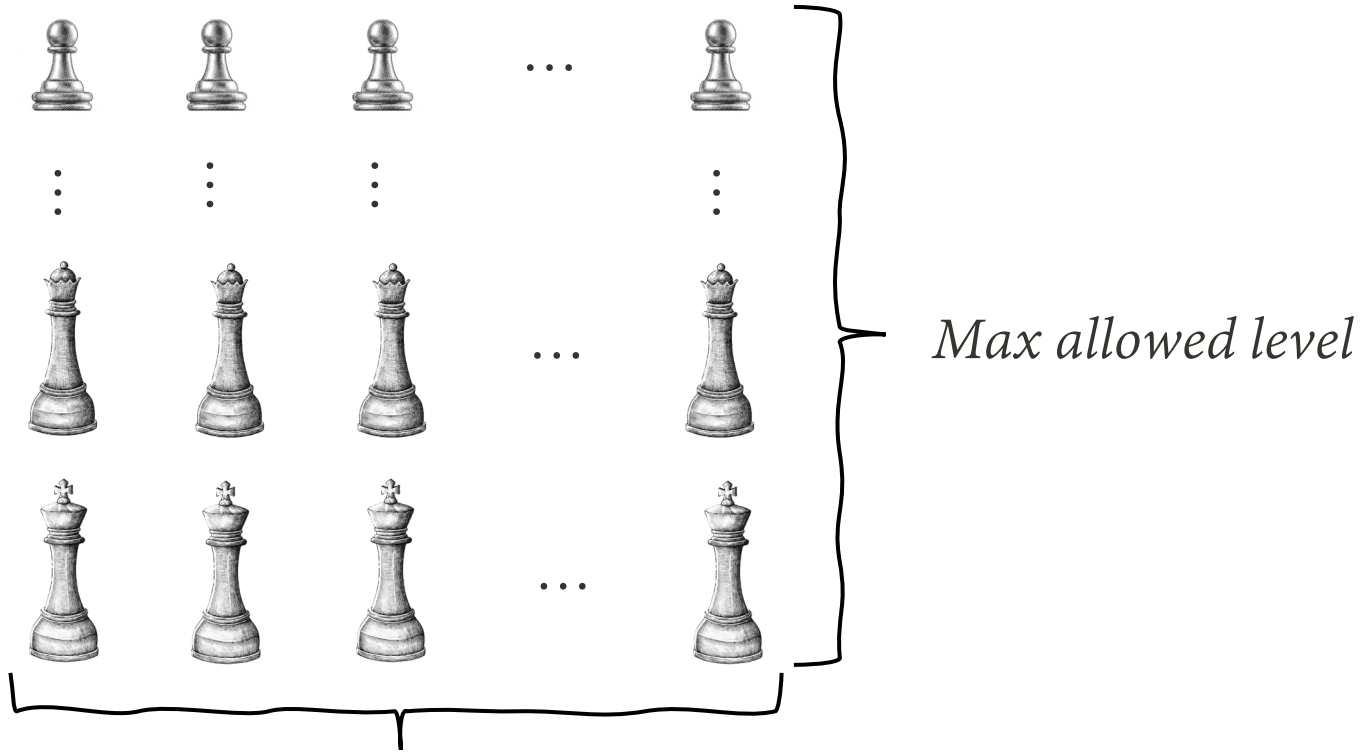
650

Charles

85

700

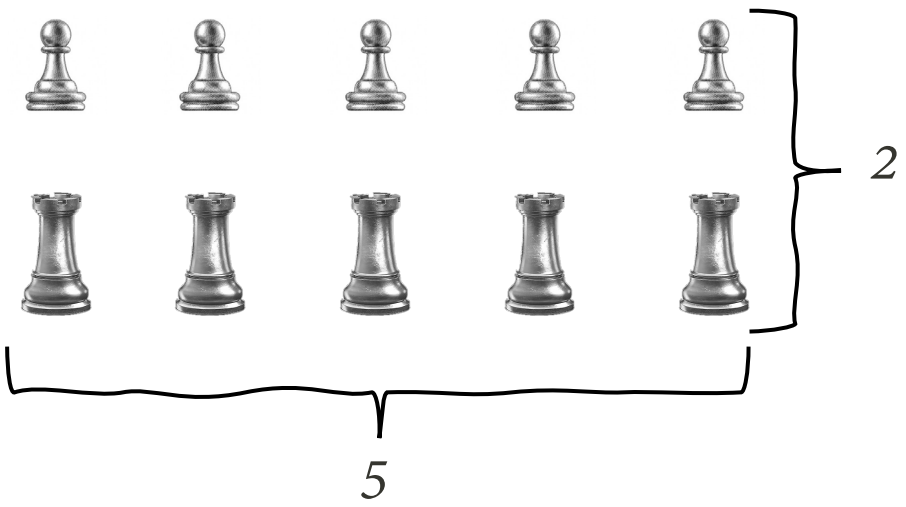
Selection search space



Selection search space (Ex #1)

Max allowed level = 2

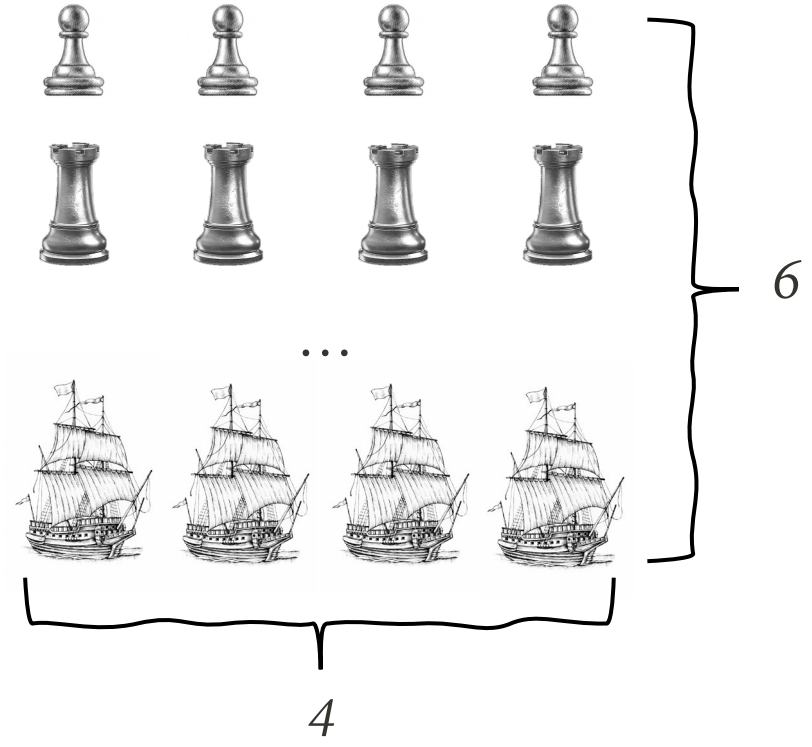
Number of available pieces per level = 5



Selection search space (Ex #2)

Max allowed level = 6

Number of available pieces per level = 4



Design

- **Trial #1**

  User → Dynamic programming Computer → Greedy approach

- **Trial #2**

  User → Dynamic programming Computer → Randomized approach

Design

```
1  int GOLD_AMOUNT;
2  int MAX_LEVEL_ALLOWED;
3  int NUMBER_OF_AVAILABLE_PIECES_PER_LEVEL;
```

Code 1: *Required user inputs*

For each algorithm print the following:

- How much gold spend?
- What is the total attack points of your army?
- What is the execution time of the algorithm?

```
// Listing the selected heroes
Trial #1
My algorithm (Dynamic programming):
        Pawn Wymond (Pawn, 60 Gold,  595 Attack)
        Castel del Monte (Castle, … )
        Philip II Augustus (Knight, …)
        …

Computer (Greedy approach):
        …

Trial #2
My algorithm (Dynamic programming):
        …

Computer (Randomized approach):
        …
```