

# Algoritmi di ordinamento (*sorting algorithms*)

**Liceo G.B. Brocchi - Bassano del Grappa (VI)**  
**Liceo Scientifico - opzione scienze applicate**  
Giovanni Mazzocchin

# Motivazioni

- Il nostro obiettivo è **ordinare** una struttura dati sulla base del valore di una chiave e del significato dell'operatore *confronto*:
  - ad esempio, per gli interi sappiamo che è definito l'ordinamento:  $5 < 6, 8 < 9$ , etc...
  - altro esempio: per le stringhe di caratteri di un alfabeto è definito l'ordinamento *lessicografico* (l'ordine del dizionario):
    - aceto < acqua < birra < gingerino < grappa < vino < vodka
  - per ora proveremo ad ordinare soltanto array di interi
- Questo problema ha innumerevoli applicazioni in Informatica:
  - ordinare per poi cercare una chiave (pensate alla ricerca binaria)
  - produrre output ordinati da input disordinati
  - fare statistiche su insiemi di dati (pensare ad esempio alla *mediana*)

# Bogo sort

- Ci sono diversi metodi (che si tradurranno in *algoritmi*) per ordinare una sequenza di interi
- Alcuni metodi sono intuitivi ma poco efficienti, altri sono meno intuitivi ma molto più efficienti
- **Bogo sort** è un algoritmo che mostra come sia effettivamente possibile ordinare un array in un modo molto intuitivo, ma spaventosamente lento. Lo pseudocodice di questo algoritmo è il seguente:

```
int v[10] = {0, 4, 7, 2, 8, 6, 2, 3, 10, 9};  
bool sorted = false;  
while (sorted == false) {  
    if (is_sorted(v)) {  
        sorted = true;  
    }  
    shuffle(v);  
}
```

# Bogo sort

- Ordinare un array con **Bogo sort** equivale a ordinare un mazzo di carte lanciandolo in aria diverse volte finché non cade ordinato
- Questo algoritmo esegue un numero di operazioni proporzionale al **fattoriale** della dimensione dell'array. In Informatica, quando sentite parlare di fattoriali ed esponenziali dovete spaventarvi...

# Insertion sort

- Considerazioni intuitive:
  - se un array  $A$  è ordinato in senso crescente, allora  $A[i] \geq A[j]$  per ogni  $i, j$  indici dell'array, con  $j < i$
  - in altre parole: se un array è ordinato in senso crescente, ogni elemento è  $\geq$  uguale di tutti i precedenti
- Sfruttiamo questa intuizione per ordinare un array
- Leggiamo l'array da sinistra a destra, controllando come è posizionato ciascun elemento rispetto ai precedenti

# Insertion sort

10	11	14	8	20	19	7	21	13	11	-1

--	--	--	--	--	--	--	--	--	--	--

- si inizia a leggere dal secondo elemento perché il primo non ha elementi precedenti
- 11 è posizionato bene rispetto ai precedenti. Nessuna operazione

# Insertion sort

<b>10</b>	<b>11</b>	<b>14</b>	<b>8</b>	<b>20</b>	<b>19</b>	<b>7</b>	<b>21</b>	<b>13</b>	<b>11</b>	<b>-1</b>

--	--	--	--	--	--	--	--	--	--	--

- 14 è posizionato bene rispetto ai precedenti. Nessuna operazione

# Insertion sort

<b>10</b>	<b>11</b>	<b>14</b>	<b>8</b>	<b>20</b>	<b>19</b>	<b>7</b>	<b>21</b>	<b>13</b>	<b>11</b>	<b>-1</b>

--	--	--	--	--	--	--	--	--	--	--

- 8 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 8$
- tutti gli elementi  $> 8$  che lo precedono vanno shiftati a destra di una posizione



# Insertion sort

<b>10</b>	<b>11</b>	<b>14</b>		<b>20</b>	<b>19</b>	<b>7</b>	<b>21</b>	<b>13</b>	<b>11</b>	<b>-1</b>

			<b>8</b>							
--	--	--	----------	--	--	--	--	--	--	--

- 8 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 8$
- tutti gli elementi  $> 8$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>10</b>	<b>11</b>		<b>14</b>	<b>20</b>	<b>19</b>	<b>7</b>	<b>21</b>	<b>13</b>	<b>11</b>	<b>-1</b>

			<b>8</b>							
--	--	--	----------	--	--	--	--	--	--	--

- 8 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 8$
- tutti gli elementi  $> 8$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>10</b>		<b>11</b>	<b>14</b>	<b>20</b>	<b>19</b>	<b>7</b>	<b>21</b>	<b>13</b>	<b>11</b>	<b>-1</b>

			<b>8</b>							
--	--	--	----------	--	--	--	--	--	--	--

- 8 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 8$
- tutti gli elementi  $> 8$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

	10	11	14	20	19	7	21	13	11	-1

			8							
--	--	--	---	--	--	--	--	--	--	--

- 8 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 8$
- tutti gli elementi  $> 8$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

8	10	11	14	20	19	7	21	13	11	-1

--	--	--	--	--	--	--	--	--	--	--

- 8 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 8$
- tutti gli elementi  $> 8$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>8</b>	<b>10</b>	<b>11</b>	<b>14</b>	<b>20</b>	<b>19</b>	<b>7</b>	<b>21</b>	<b>13</b>	<b>11</b>	<b>-1</b>

--	--	--	--	--	--	--	--	--	--	--

- 20 è posizionato bene rispetto ai precedenti. Nessuna operazione

# Insertion sort

<b>8</b>	<b>10</b>	<b>11</b>	<b>14</b>	<b>20</b>	<b>19</b>	<b>7</b>	<b>21</b>	<b>13</b>	<b>11</b>	<b>-1</b>

--	--	--	--	--	--	--	--	--	--	--

- 19 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 19$
- tutti gli elementi  $> 19$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

8	10	11	14	20		7	21	13	11	-1

					19					
--	--	--	--	--	----	--	--	--	--	--

- 19 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 19$
- tutti gli elementi  $> 19$  che lo precedono vanno shiftati a destra di una posizione



# Insertion sort

8	10	11	14		20	7	21	13	11	-1

					19					
--	--	--	--	--	----	--	--	--	--	--

- 19 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 19$
- tutti gli elementi  $> 19$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

8	10	11	14	19	20	7	21	13	11	-1

--	--	--	--	--	--	--	--	--	--	--

- 19 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 19$
- tutti gli elementi  $> 19$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>8</b>	<b>10</b>	<b>11</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>7</b>	<b>21</b>	<b>13</b>	<b>11</b>	<b>-1</b>

--	--	--	--	--	--	--	--	--	--	--

- 7 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 7$
- tutti gli elementi  $> 7$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

8	10	11	14	19	20		21	13	11	-1

						7				
--	--	--	--	--	--	---	--	--	--	--

- 7 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 7$
- tutti gli elementi  $> 7$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

8	10	11	14	19		20	21	13	11	-1

						7				
--	--	--	--	--	--	---	--	--	--	--

- 7 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 7$
- tutti gli elementi  $> 7$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

8	10	11	14		19	20	21	13	11	-1

  

						7				
--	--	--	--	--	--	---	--	--	--	--

- 7 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 7$
- tutti gli elementi  $> 7$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

8	10	11		14	19	20	21	13	11	-1

						7				
--	--	--	--	--	--	---	--	--	--	--

- 7 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 7$
- tutti gli elementi  $> 7$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

8	10		11	14	19	20	21	13	11	-1

						7				
--	--	--	--	--	--	---	--	--	--	--

- 7 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 7$
- tutti gli elementi  $> 7$  che lo precedono vanno shiftati a destra di una posizione



# Insertion sort

8		10	11	14	19	20	21	13	11	-1

						7				
--	--	--	--	--	--	---	--	--	--	--

- 7 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 7$
- tutti gli elementi  $> 7$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

	8	10	11	14	19	20	21	13	11	-1

						7				
--	--	--	--	--	--	---	--	--	--	--

- 7 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 7$
- tutti gli elementi  $> 7$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>13</b>	<b>11</b>	<b>-1</b>

--	--	--	--	--	--	--	--	--	--	--

- 7 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 7$
- tutti gli elementi  $> 7$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>13</b>	<b>11</b>	<b>-1</b>

--	--	--	--	--	--	--	--	--	--	--

- 21 è posizionato bene rispetto ai precedenti. Nessuna operazione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>13</b>	<b>11</b>	<b>-1</b>

--	--	--	--	--	--	--	--	--	--	--

- 13 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 13$
- tutti gli elementi  $> 13$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>		<b>11</b>	<b>-1</b>

								<b>13</b>		
--	--	--	--	--	--	--	--	-----------	--	--

- 13 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 13$
- tutti gli elementi  $> 13$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>14</b>	<b>19</b>	<b>20</b>		<b>21</b>	<b>11</b>	<b>-1</b>

								<b>13</b>		
--	--	--	--	--	--	--	--	-----------	--	--

- 13 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 13$
- tutti gli elementi  $> 13$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>14</b>	<b>19</b>		<b>20</b>	<b>21</b>	<b>11</b>	<b>-1</b>

								<b>13</b>		
--	--	--	--	--	--	--	--	-----------	--	--

- 13 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 13$
- tutti gli elementi  $> 13$  che lo precedono vanno shiftati a destra di una posizione



# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>14</b>		<b>19</b>	<b>20</b>	<b>21</b>	<b>11</b>	<b>-1</b>

								<b>13</b>		
--	--	--	--	--	--	--	--	-----------	--	--

- 13 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 13$
- tutti gli elementi  $> 13$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>		<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>11</b>	<b>-1</b>

								<b>13</b>		
--	--	--	--	--	--	--	--	-----------	--	--

- 13 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 13$
- tutti gli elementi  $> 13$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>13</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>11</b>	<b>-1</b>

--	--	--	--	--	--	--	--	--	--	--

- 13 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 13$
- tutti gli elementi  $> 13$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>13</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>11</b>	<b>-1</b>

--	--	--	--	--	--	--	--	--	--	--

- 11 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 11$
- tutti gli elementi  $> 11$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>13</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>		<b>-1</b>

									<b>11</b>	
--	--	--	--	--	--	--	--	--	-----------	--

- 11 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 11$
- tutti gli elementi  $> 11$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>13</b>	<b>14</b>	<b>19</b>	<b>20</b>		<b>21</b>	<b>-1</b>

									<b>11</b>	
--	--	--	--	--	--	--	--	--	-----------	--

- 11 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 11$
- tutti gli elementi  $> 11$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>13</b>	<b>14</b>	<b>19</b>		<b>20</b>	<b>21</b>	<b>-1</b>

									<b>11</b>	
--	--	--	--	--	--	--	--	--	-----------	--

- 11 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 11$
- tutti gli elementi  $> 11$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>13</b>	<b>14</b>		<b>19</b>	<b>20</b>	<b>21</b>	<b>-1</b>

									<b>11</b>	
--	--	--	--	--	--	--	--	--	-----------	--

- 11 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 11$
- tutti gli elementi  $> 11$  che lo precedono vanno shiftati a destra di una posizione



# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>13</b>		<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>-1</b>

									<b>11</b>	
--	--	--	--	--	--	--	--	--	-----------	--

- 11 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 11$
- tutti gli elementi  $> 11$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>		<b>13</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>-1</b>

									<b>11</b>	
--	--	--	--	--	--	--	--	--	-----------	--

- 11 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 11$
- tutti gli elementi  $> 11$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>11</b>	<b>13</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>-1</b>

--	--	--	--	--	--	--	--	--	--	--

- 11 è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq 11$
- tutti gli elementi  $> 11$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>11</b>	<b>13</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>-1</b>

--	--	--	--	--	--	--	--	--	--	--

- $-1$  è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq -1$
- tutti gli elementi  $> -1$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>11</b>	<b>13</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>	

										<b>-1</b>
--	--	--	--	--	--	--	--	--	--	-----------

- $-1$  è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq -1$
- tutti gli elementi  $> -1$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>11</b>	<b>13</b>	<b>14</b>	<b>19</b>	<b>20</b>		<b>21</b>

										<b>-1</b>
--	--	--	--	--	--	--	--	--	--	-----------

- $-1$  è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq -1$
- tutti gli elementi  $> -1$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

7	8	10	11	11	13	14	19		20	21

										-1
--	--	--	--	--	--	--	--	--	--	----

- $-1$  è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq -1$
- tutti gli elementi  $> -1$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>11</b>	<b>13</b>	<b>14</b>		<b>19</b>	<b>20</b>	<b>21</b>

										<b>-1</b>
--	--	--	--	--	--	--	--	--	--	-----------

- $-1$  è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq -1$
- tutti gli elementi  $> -1$  che lo precedono vanno shiftati a destra di una posizione



# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>11</b>	<b>13</b>		<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>

										<b>-1</b>
--	--	--	--	--	--	--	--	--	--	-----------

- $-1$  è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq -1$
- tutti gli elementi  $> -1$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>11</b>		<b>13</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>

										<b>-1</b>
--	--	--	--	--	--	--	--	--	--	-----------

- $-1$  è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq -1$
- tutti gli elementi  $> -1$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>		<b>11</b>	<b>13</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>

										<b>-1</b>
--	--	--	--	--	--	--	--	--	--	-----------

- $-1$  è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq -1$
- tutti gli elementi  $> -1$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>	<b>10</b>		<b>11</b>	<b>11</b>	<b>13</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>

										<b>-1</b>
--	--	--	--	--	--	--	--	--	--	-----------

- $-1$  è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq -1$
- tutti gli elementi  $> -1$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>	<b>8</b>		<b>10</b>	<b>11</b>	<b>11</b>	<b>13</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>

										<b>-1</b>
--	--	--	--	--	--	--	--	--	--	-----------

- $-1$  è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq -1$
- tutti gli elementi  $> -1$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

<b>7</b>		<b>8</b>	<b>10</b>	<b>11</b>	<b>11</b>	<b>13</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>

										<b>-1</b>
--	--	--	--	--	--	--	--	--	--	-----------

- $-1$  è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq -1$
- tutti gli elementi  $> -1$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

	<b>7</b>	<b>8</b>	<b>10</b>	<b>11</b>	<b>11</b>	<b>13</b>	<b>14</b>	<b>19</b>	<b>20</b>	<b>21</b>

										<b>-1</b>
--	--	--	--	--	--	--	--	--	--	-----------

- $-1$  è posizionato male rispetto ai precedenti. Va sistemato al posto giusto, ossia appena dopo il primo elemento precedente  $\leq -1$
- tutti gli elementi  $> -1$  che lo precedono vanno shiftati a destra di una posizione

# Insertion sort

-1	7	8	10	11	11	13	14	19	20	21

--	--	--	--	--	--	--	--	--	--	--

- l'array è ordinato
- è un ordinamento di tipo **incrementale**: alla fine dell'iterazione  $i$ , la fetta dell'array da 0 a  $i$  è ordinata
- quando si verifica il **worst case** di Insertion sort?
- quando si verifica il **best case** di Insertion sort?



# Insertion sort

```
for i = 1, i <= A.size - 1, i = i + 1:  
    item = A[i]  
    j = i - 1  
    while j >= 0 and item < A[j]:  
        A[j + 1] = A[j]  
        j = j - 1  
    A[j + 1] = item
```

- l'assegnazione  $A[j + 1] = A[j]$  realizza il right shift
- perché  $A[j + 1] = \text{item}$ ? Se  $j$  è ultimo indice per il quale  $\text{item} < A[j]$ , allora  $\text{item}$  va posto in posizione  $j$ . Ma, dato che  $\text{item} < A[j]$ , si entra nel corpo del ciclo, al termine del quale  $j$  viene decrementata di 1. Quindi, quando si esce dal ciclo, la posizione giusta in cui porre  $\text{item}$  è  $j + 1$

# Bubble sort

- Considerazione intuitiva:
  - se un array  $A$  è ordinato in senso crescente, allora è vero che  $A[i] \leq A[i + 1]$  per ogni  $i$  indice dell'array
- Quindi, se un array non è ordinato in senso crescente, allora esiste almeno un  $i$  per il quale  $A[i] > A[i + 1]$
- **Bubble sort** esamina le coppie  $(A[i], A[i + 1])$  una ad una, sistemando quelle disordinate
- Bubble sort procede per *passate*, vediamo come

# Bubble sort

1st pass										
swap	6	3	6	87	11	13	6	3	9	12
	3	6	6	87	11	13	6	3	9	12
	3	6	6	87	11	13	6	3	9	12
swap	3	6	6	87	11	13	6	3	9	12
swap	3	6	6	11	87	13	6	3	9	12
swap	3	6	6	11	13	87	6	3	9	12
swap	3	6	6	11	13	6	87	3	9	12
swap	3	6	6	11	13	6	3	87	9	12
swap	3	6	6	11	13	6	3	9	87	12
	3	6	6	11	13	6	3	9	12	87

2nd pass										
swap swap swap swap	3	6	6	11	13	6	3	9	12	87
	3	6	6	11	13	6	3	9	12	87
	3	6	6	11	13	6	3	9	12	87
	3	6	6	11	13	6	3	9	12	87
	3	6	6	11	13	6	3	9	12	87
	3	6	6	11	6	13	3	9	12	87
	3	6	6	11	6	3	13	9	12	87
	3	6	6	11	6	3	9	13	12	87
	3	6	6	11	6	3	9	12	13	87

- in grigio le coppie esaminate ed eventualmente sistemate
- gli elementi blu sono sistemati correttamente, per cui non si sposteranno più
- in verde la porzione dell'array non ancora sistemata al termine della passata

# Bubble sort

3rd pass

	3	6	6	11	6	3	9	12	13	87
	3	6	6	11	6	3	9	12	13	87
	3	6	6	11	6	3	9	12	13	87
swap	3	6	6	11	6	3	9	12	13	87
swap	3	6	6	6	11	3	9	12	13	87
swap	3	6	6	6	3	11	9	12	13	87
	3	6	6	6	3	9	11	12	13	87
	3	6	6	6	3	9	11	12	13	87

4th pass

	3	6	6	6	3	9	11	12	13	87
	3	6	6	6	3	9	11	12	13	87
	3	6	6	6	3	9	11	12	13	87
swap	3	6	6	6	3	9	11	12	13	87
	3	6	6	3	6	9	11	12	13	87
	3	6	6	3	6	9	11	12	13	87
	3	6	6	3	6	9	11	12	13	87

5th pass

	3	6	6	3	6	9	11	12	13	87
	3	6	6	3	6	9	11	12	13	87
swap	3	6	6	3	6	9	11	12	13	87
	3	6	3	6	6	9	11	12	13	87
	3	6	3	6	6	9	11	12	13	87
	3	6	3	6	6	9	11	12	13	87

# Bubble sort

		6th pass									
swap		3	6	3	6	6	9	11	12	13	87
		3	6	3	6	6	9	11	12	13	87
		3	3	6	6	6	9	11	12	13	87
		3	3	6	6	6	9	11	12	13	87
		3	3	6	6	6	9	11	12	13	87

7th pass									
3	3	6	6	6	9	11	12	13	87
3	3	6	6	6	9	11	12	13	87
3	3	6	6	6	9	11	12	13	87
3	3	6	6	6	9	11	12	13	87

8th pass									
3	3	6	6	6	9	11	12	13	87
3	3	6	6	6	9	11	12	13	87
3	3	6	6	6	9	11	12	13	87

9th pass									
3	3	6	6	6	9	11	12	13	87
3	3	6	6	6	9	11	12	13	87

# Bubble sort

- Cosa c'entrano le **bolle** con l'ordinamento?
- Bubble sort deve il suo nome al fatto che, al termine di ciascuna passata, l'elemento più grande non ancora sistemato (la bolla più grande che non è ancora salita) si sistema nella prima posizione libera a destra (va più in alto possibile)
- Questa è anche una dimostrazione intuitiva della **correttezza** dell'algoritmo: se dopo ciascuna passata viene posizionato correttamente l'elemento più grande che doveva essere sistemato, dopo massimo  $n - 1$  (dove  $n$  è la dimensione dell'array) passate l'array è completamente ordinato
- Ottimizzazione: se al termine di una passata si verifica che tutte le coppie considerate sono ordinate, allora non ha senso fare ulteriori passate

# Bubble sort

```
for i = A.size - 1, i >= 1, i = i - 1:  
    for j = 0, j <= i - 1, j = j + 1:  
        if A[j] > A[j + 1]:  
            swap(A[j], A[j + 1])
```

- il ciclo esterno procede da destra a sinistra perché Bubble sort sistema gli elementi giusti al posto giusto a partire da destra
- il ciclo interno arriva fino a  $i - 1$  perché nel corpo del ciclo ciascun elemento viene confrontato con il successore. L'ultimo successore con cui ha senso effettuare il confronto è l'elemento in posizione  $i$ , il cui predecessore è in posizione  $i - 1$
- al termine della passata di indice  $i$ , in posizione  $i$  c'è l'elemento corretto, che di conseguenza non verrà più spostato

# Selection sort

- Considerazione intuitiva:
  - l'elemento più piccolo di un array ordinato in senso crescente deve essere posizionato all'inizio dell'array
- Sfruttiamo la considerazione precedente per studiare un algoritmo di ordinamento molto semplice e inefficiente:  
**Selection sort**
- Per implementare questo algoritmo, dobbiamo ricordarci come si cerca l'elemento minimo di un array, e come si scambia il contenuto di 2 variabili, cose che sappiamo già fare



# Selection sort

indexes	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
values	<b>9</b>	<b>3</b>	<b>10</b>	<b>2</b>	<b>4</b>	<b>20</b>	<b>10</b>	<b>11</b>	<b>9</b>	<b>6</b>

## **iterazione 1:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

indexes	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
values	<b>2</b>	<b>3</b>	<b>10</b>	<b>9</b>	<b>4</b>	<b>20</b>	<b>10</b>	<b>11</b>	<b>9</b>	<b>6</b>

## **iterazione 1:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>10</b>	<b>9</b>	<b>4</b>	<b>20</b>	<b>10</b>	<b>11</b>	<b>9</b>	<b>6</b>

**iterazione 2:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>10</b>	<b>9</b>	<b>4</b>	<b>20</b>	<b>10</b>	<b>11</b>	<b>9</b>	<b>6</b>

## **iterazione 2:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>10</b>	<b>9</b>	<b>4</b>	<b>20</b>	<b>10</b>	<b>11</b>	<b>9</b>	<b>6</b>

## **iterazione 3:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>9</b>	<b>10</b>	<b>20</b>	<b>10</b>	<b>11</b>	<b>9</b>	<b>6</b>

## **iterazione 3:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>9</b>	<b>10</b>	<b>20</b>	<b>10</b>	<b>11</b>	<b>9</b>	<b>6</b>

## **iterazione 4:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>6</b>	<b>10</b>	<b>20</b>	<b>10</b>	<b>11</b>	<b>9</b>	<b>9</b>

## **iterazione 4:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati



# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>6</b>	<b>10</b>	<b>20</b>	<b>10</b>	<b>11</b>	<b>9</b>	<b>9</b>

## **iterazione 5:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>6</b>	<b>9</b>	<b>20</b>	<b>10</b>	<b>11</b>	<b>10</b>	<b>9</b>

## **iterazione 5:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>6</b>	<b>9</b>	<b>20</b>	<b>10</b>	<b>11</b>	<b>10</b>	<b>9</b>

## **iterazione 6:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>6</b>	<b>9</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>10</b>	<b>20</b>

## **iterazione 6:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>6</b>	<b>9</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>10</b>	<b>20</b>

## **iterazione 7:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>6</b>	<b>9</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>10</b>	<b>20</b>

## **iterazione 7:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>6</b>	<b>9</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>10</b>	<b>20</b>

## **iterazione 8:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>6</b>	<b>9</b>	<b>9</b>	<b>10</b>	<b>10</b>	<b>11</b>	<b>20</b>

## **iterazione 8:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati



# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>6</b>	<b>9</b>	<b>9</b>	<b>10</b>	<b>10</b>	<b>11</b>	<b>20</b>

## **iterazione 9:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>6</b>	<b>9</b>	<b>9</b>	<b>10</b>	<b>10</b>	<b>11</b>	<b>20</b>

## **iterazione 9:**

trovare l'elemento minimo della porzione grigia  
sistemarlo all'inizio della porzione grigia

gli elementi evidenziati in viola sono stati swappati

# Selection sort

<b>indexes</b>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<b>values</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>6</b>	<b>9</b>	<b>9</b>	<b>10</b>	<b>10</b>	<b>11</b>	<b>20</b>

- la porzione rimasta ha un solo elemento, che ovviamente è già posizionato bene
- l'array è ordinato
- l'efficienza di Selection sort dipende dagli elementi dell'array?

# Selection sort

```
for i = 0, i <= A.size - 2, i = i + 1:  
    min = A[i]  
    ind_min = i  
    for j = i + 1, j <= A.size - 1, j = j + 1:  
        if A[j] < min:  
            min = A[j]  
            ind_min = j  
    swap(A[i], A[ind_min])
```

# Considerazioni

- L'ordinamento è uno dei problemi più studiati dell'Informatica
- Gli algoritmi che abbiamo visto sono detti **in place**: operano senza utilizzare memoria aggiuntiva rispetto a quella necessaria per l'array
- Vedremo altri algoritmi di ordinamento meno intuitivi, ma molto più efficienti

# Da vedere a casa

- [Programming BASIC and Sorting - Computerphile](#)