Università degli Studi di Cassino e del Lazio Meridionale

Corso di Fondamenti di Informatica

Algoritmi su array /1

Anno Accademico 2016/2017

Francesco Tortorella

Algoritmi su array

- Quando si usano gli array, si eseguono frequentemente alcune operazioni “tipiche”:
  - inizializzazione
  - lettura
  - stampa
  - ricerca del minimo e del massimo
  - ricerca di un valore
  - eliminazione di un valore
  - inserimento di un valore
  - ordinamento del vettore

Ricerca di un valore nell'array

Si voglia verificare se in un array è presente almeno un'istanza di un certo valore. In caso positivo si fornisca l'indice dell'elemento.

- Non è una ricerca esaustiva…
- Quali costrutti usare ?
- Come gestire l'esito della ricerca ?

Ricerca sequenziale in array

```
int cerca_fun(int vet[], int n, int x) {
    int i, pos;

    i = 0;
    pos = -1;

    while (i < n && pos < 0 ) {
        if (vet[i] == x)
            pos = i;
        else i++;
    }

    return (pos);
}
```

Chiamata nel main

```
pos =cerca_fun(vet,n,x);
if(pos>=0)
  cout <<"OK: " << pos << endl;
else
  cout << "Non trovato\n" ;
```

Ricerca sequenziale in array

```
int cerca_fun(int vet[], int n, int x) {
    int i, pos;

    i = 0;
    pos = -1;

    while (i < n && pos < 0 ) {
        if (vet[i] == x)
            pos = i;
        else i++;
    }

    return (pos);
}
```

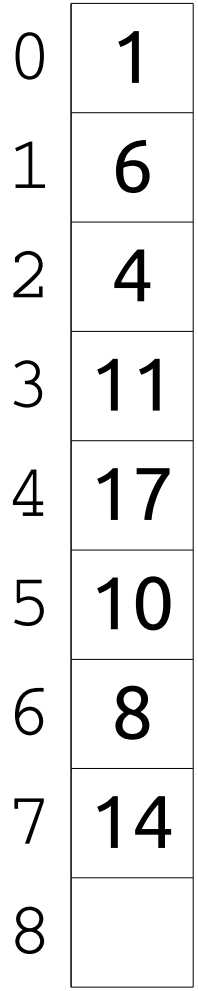
Valutazione short circuit

Ricerca sequenziale in array: uso della sentinella

- E' possibile eliminare il controllo sull'indice se si inserisce in coda all'array un ulteriore elemento (*sentinella*) che ha valore uguale a quello cercato.
- In questo modo, la ricerca ha sicuramente successo.
- Dobbiamo però verificare se l'istanza trovata al termine del ciclo è reale o fittizia.

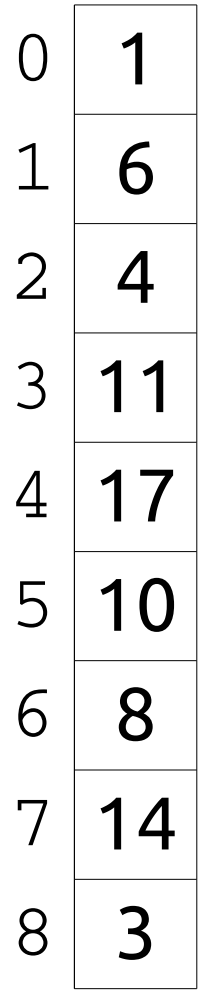
Ricerca sequenziale in array: uso della sentinella

Array originale
n=8
x=3



Spazio disponibile

Array modificato con l'inserimento della sentinella



Ricerca sequenziale in array: uso della sentinella

```
int cerca_fun_sentinella(int vet[], int n, int x) {
    int i, pos;

    i = 0;
    pos = -1;

    while (pos < 0 ) {
        if (vet[i] == x)
            pos = i;
        else i++;
    }

    if (pos < n)
      return (pos);
    else
      return (-1);
}
```

Ho trovato la sentinella ?

Ricerca sequenziale in array: uso della sentinella

- La ricerca sequenziale con sentinella permette di semplificare la condizione di terminazione del ciclo, ma è possibile solo se si è certi che c'è spazio nell'array.

Problema: Conteggio numero di occorrenze di un valore nell'array

- Si voglia verificare se in un array è presente un certo valore. In caso positivo si fornisca il numero e le posizioni delle istanze presenti.
- Questa volta si tratta di una ricerca esaustiva.

Ricerca posizioni delle occorrenze di un valore nell'array

```
void cerca_occ(int v[], int n, int x, int pos[], int &nocc){
    int i;

    nocc = 0;

    for(i = 0; i < n; i++)
        if(vet[i] == x)
            pos[nocc++] = i;

    return;
}
```

Problema: Ricerca posizioni delle occorrenze del minimo nell'array

- Si voglia cercare il valore minimo in un array insieme con le posizioni delle varie occorrenze.
- Anche stavolta è una ricerca esaustiva
- Achtung: il valore da considerare cambia durante la scansione dell'array …

Array ordinati

- Un array si dice ordinato se i suoi elementi sono disposti in modo da verificare una relazione d'ordine, per cui ogni elemento è minore (o maggiore) di quello che lo segue.
- In questi casi, le operazioni di ricerca possono essere realizzate in modo più efficiente (es. ricerca del minimo).

Ricerca di un valore in un array ordinato

Si voglia verificare se in un array ordinato è presente almeno un'istanza di un certo valore. In caso positivo si fornisca l'indice dell'elemento.

- Quali costrutti usare ?
- Come gestire l'esito della ricerca ?
- Come sfruttare l'ordinamento dell'array ?

Ricerca di un valore in un array ordinato (in senso crescente)

```
int cerca_ord(int vet[], int n, int x) {
    int i, pos;

    i = 0;
    pos = -1;

    while (i < n && x >= vet[i] && pos < 0)
        if (vet[i] == x)
            pos = i;
        else i++;
    return (pos);
}
```

Ricerca di un valore in un array ordinato (in senso crescente)

```
int cerca_ord(int vet[], int n, int x) {
    int i, pos;

    i = 0;
    pos = -1;

    while (i < n && x >= vet[i] && pos < 0)
        if (vet[i] == x)
            pos = i;
        else i++;
    return (pos);
}
```

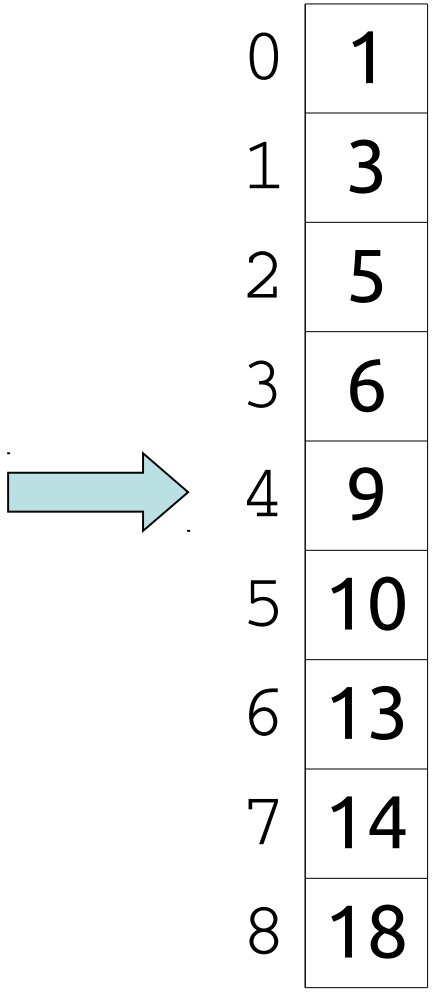
Valutazione short circuit

Ricerca di un valore in un array ordinato

- Quanti confronti si eseguono con l'algoritmo appena visto ?
- E' possibile organizzare opportunamente la ricerca per ridurre i confronti ?

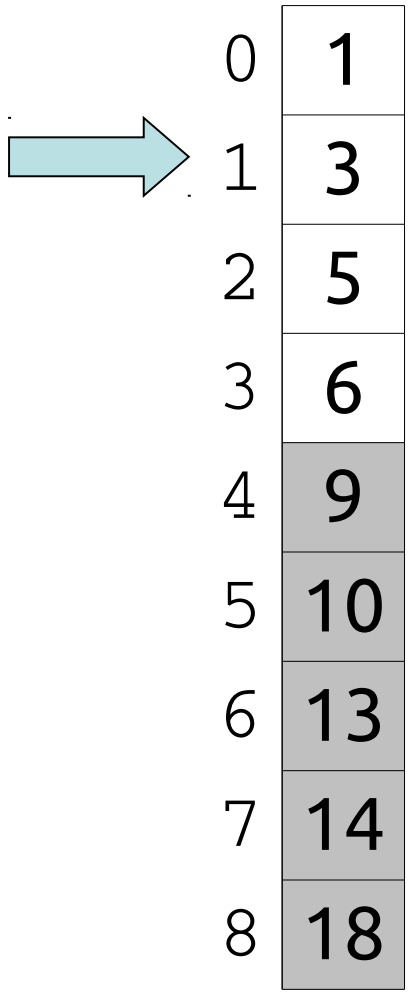
Ricerca di un valore in un array ordinato

- Si supponga di dover cercare il valore 5
- Si confronti inizialmente il valore con l'elemento centrale
- Siccome 5<9 possiamo trascurare la seconda metà dell'array perché sicuramente non conterrà il valore cercato



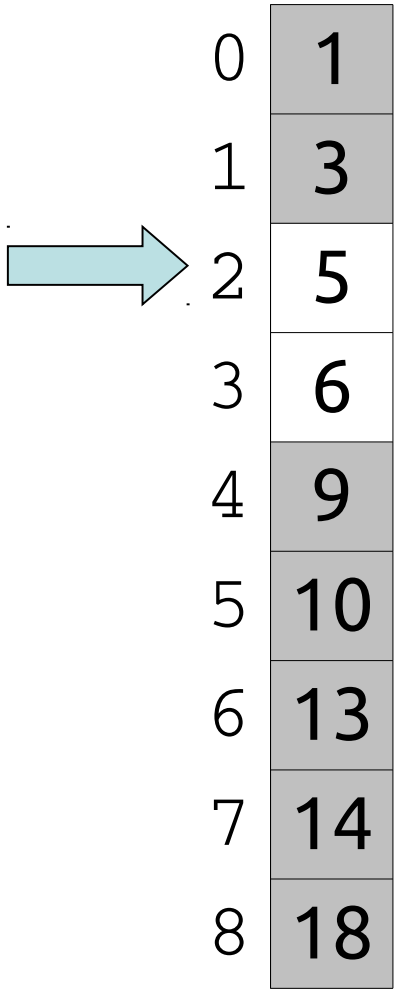
Ricerca di un valore in un array ordinato

- Concentriamo l'attenzione sulla parte superiore e nuovamente confrontiamo il valore con l'elemento centrale della parte "superstite" dell'array
- Siccome 5>3, possiamo trascurare la metà superiore della parte di array che stiamo considerando



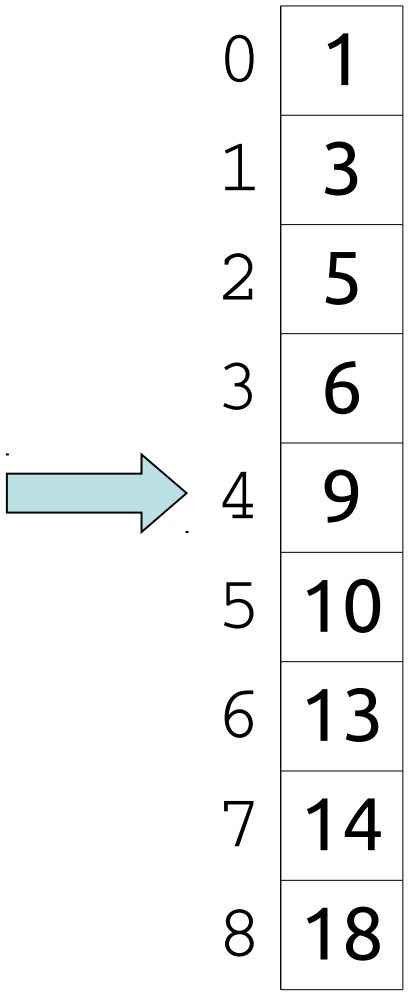
Ricerca di un valore in un array ordinato

- Questa volta l'elemento centrale della parte di array in esame è uguale a 5, per cui si arresta la scansione.
- Sono stati sufficienti 3 confronti per terminare la ricerca.



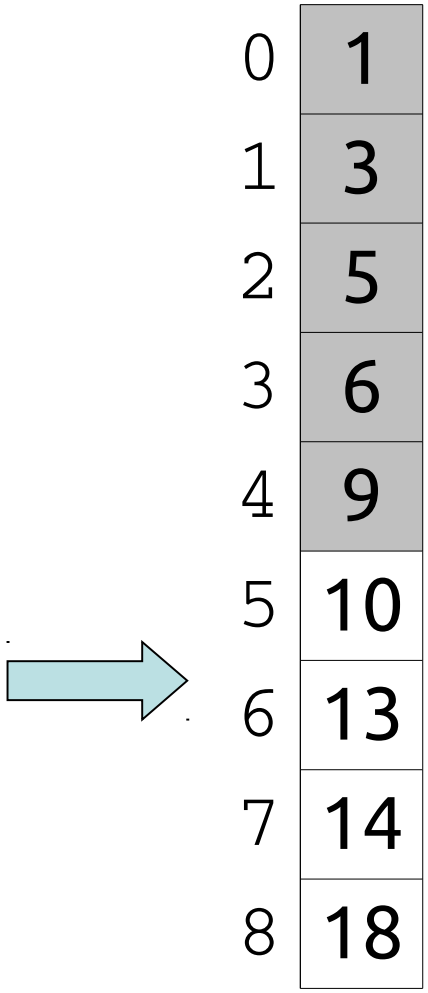
Ricerca di un valore in un array ordinato

- Si supponga ora di dover cercare il valore 15
- Si confronti inizialmente il valore con l'elemento centrale
- Siccome 15>9 possiamo trascurare la prima metà dell'array perché sicuramente non conterrà il valore cercato



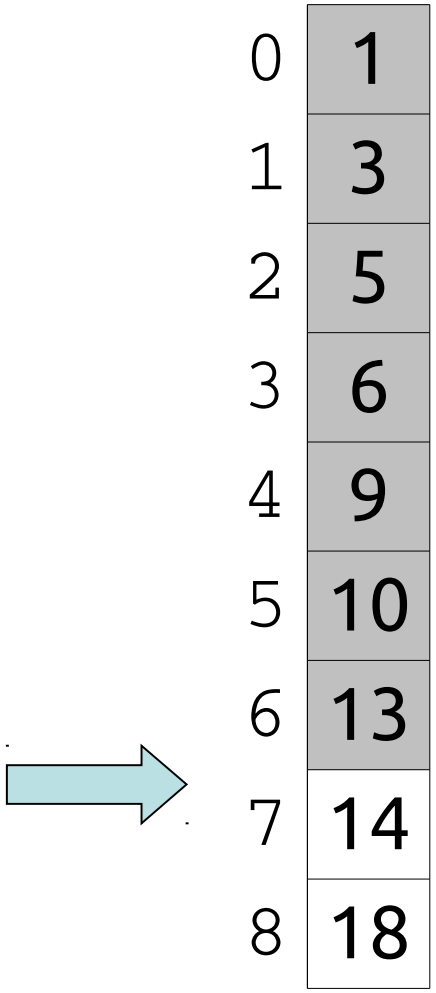
Ricerca di un valore in un array ordinato

- Concentriamo l'attenzione sulla parte inferiore e nuovamente confrontiamo il valore con l'elemento centrale della parte "superstite" dell'array
- Siccome 15>13, possiamo trascurare la metà superiore della parte di array che stiamo considerando



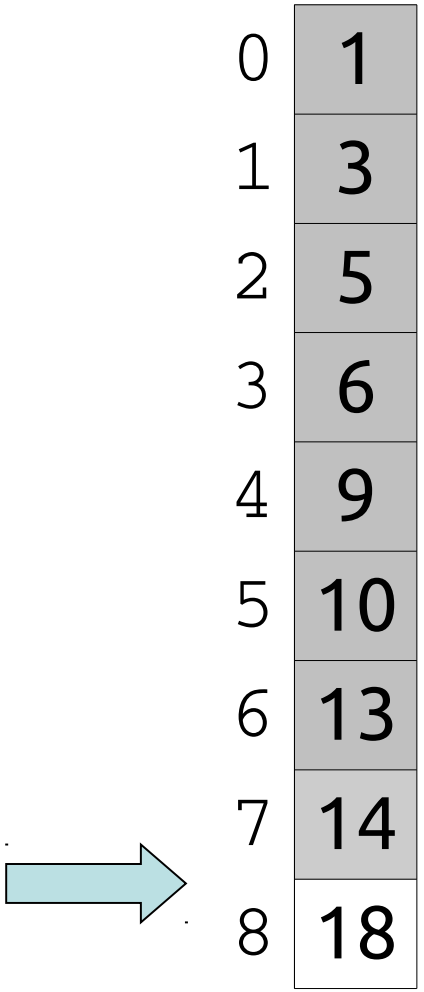
Ricerca di un valore in un array ordinato

- Anche questa volta l'elemento centrale della parte di array in esame è minore di 15, per cui si considera la parte di array successiva all'elemento centrale



Ricerca di un valore in un array ordinato

- Questa volta la parte di array è costituita da un solo elemento, diverso dal valore cercato.
- Non ci sono più altre parti di array da esplorare e quindi la ricerca è finita con esito negativo.
- Quanti confronti abbiamo fatto in tutto ?



Ricerca binaria

- L’algoritmo descritto si definisce di ricerca *binaria* o *dicotomica*.
- Come implementarlo ?
- Quali costrutti utilizzare ?

Ricerca binaria

```
int cerca_bin(int vet[], int n, int x) {
    int posin, posfin, posmed, pos;

    pos = -1;
    posin = 0;  posfin = n - 1;

    while (pos < 0 && posin <= posfin) {
        posmed = (posin + posfin) / 2;
        if (x > vet[posmed])
            posin = posmed + 1;
        else if (x < vet[posmed])
            posfin = posmed - 1;
        else
            pos = posmed;
    }

    return (pos);
}
```

Ricerca binaria

```
int cerca_bin(int vet[], int n, int x) {
    int posin, posfin, posmed, pos;

    pos = -1;
    posin = 0;  posfin = n - 1;

    while (pos < 0 && posin <= posfin) {
        posmed = (posin + posfin) / 2;
        if (x > vet[posmed])
            posin = posmed + 1;
        else if (x < vet[posmed])
            posfin = posmed - 1;
        else
            pos = posmed;
    }

    return (pos);
}
```

Che cosa indica questa condizione ?