Декларативно програмиране в процедурните езици

Множества и списъци

Владислав Ненчев

Софийски Университет “Св. Климент Охридски”
Факултет по Математика и Информатика
Катедра по Математическа Логика и Приложенията ѝ

4 Ноември 2016

Множества vs списъци

Множества	vs	Списъци

Множества vs списъци

Множества	vs	Списъци
Уникалнот на елементите.		
Изброяване: поредността няма значение.		
Операции: сечение, обединение, разлика, подмножество.		
Принадлежност и изтриване за $\mathcal{O}(n)$.		

Множества vs списъци

Множества	vs	Списъци
Уникалнот на елементите.		Повтаряне на елементите.
Изброяване: поредността няма значение.		Изброяване: поредността е важна.
Операции: сечение, обединение, разлика, подмножество.		Операции: конкатенация, zip, подсписък, подредица, пермутации, сортиране.
Принадлежност и изтриване за $\mathcal{O}(n)$.		Логаритмично или интерполационно търсене.

Множества vs списъци

Множества	vs	Списъци
Уникалнот на елементите. Изброяване: поредността няма значение. Операции: сечение, обединение, разлика, подмножество. Принадлежност и изтриване за $\mathcal{O}(n)$.		Повтаряне на елементите. Изброяване: поредността е важна. Операции: конкатенация, zip, подсписък, подредица, пермутации, сортиране. Логаритмично или интерполационно търсене.

Преобразуване на изрази за множества. Кореспонденция между множествени операции (и принадлежност) и булеви оператори.

Множества vs списъци

Множества	vs	Списъци
Уникалнот на елементите. Изброяване: поредността няма значение. Операции: сечение, обединение, разлика, подмножество. Принадлежност и изтриване за $\mathcal{O}(n)$.		Повтаряне на елементите. Изброяване: поредността е важна. Операции: конкатенация, zip, подсписък, подредица, пермутации, сортиране. Логаритмично или интерполационно търсене.

Преобразуване на изрази за множества. Кореспонденция между множествени операции (и принадлежност) и булеви оператори.

Други важни структури: map (функция), tuple (n-орка).

Множества и списъци в C#

Множества - ISet<T> и HashSet<T>:

- HashSet.Add, HashSet.Remove, HashSet.Contains
- HashSet.IntersectWith, HashSet.UnionWith, HashSet.ExceptWith, HashSet.SymmetricExceptWith
- HashSet.IsSubsetOf, HashSet.IsSupersetOf, HashSet.IsProperSubsetOf, HashSet.IsProperSupersetOf

Множества и списъци в C#

Множества - ISet<T> и HashSet<T>:

- HashSet.Add, HashSet.Remove, HashSet.Contains
- HashSet.IntersectWith, HashSet.UnionWith, HashSet.ExceptWith, HashSet.SymmetricExceptWith
- HashSet.IsSubsetOf, HashSet.IsSupersetOf, HashSet.IsProperSubsetOf, HashSet.IsProperSupersetOf

Списъци - IList <T> и List <T>:

- List .Add, List .Remove, List .Contains
- List .AddRange, List .Zip, IEnumerable.Concat
- List .Reverse, List .Sort
- List . Intersect , List .Union, List .Except

Множества и списъци в C#

Множества - ISet<T> и HashSet<T>:

- HashSet.Add, HashSet.Remove, HashSet.Contains
- HashSet.IntersectWith, HashSet.UnionWith, HashSet.ExceptWith, HashSet.SymmetricExceptWith
- HashSet.IsSubsetOf, HashSet.IsSupersetOf, HashSet.IsProperSubsetOf, HashSet.IsProperSupersetOf

Списъци - IList <T> и List <T>:

- List .Add, List .Remove, List .Contains
- List .AddRange, List .Zip, IEnumerable.Concat
- List .Reverse, List .Sort
- List . Intersect , List .Union, List .Except

Други: Hashtable<T, T>, Dictionary<T, T>, Tuple (до 8-орки)

Операции с графи

```
bool IsSubgraphOf(Graph graph) =>
  vertices.IsSubsetOf(graph.vertices) &&
    edges.IsSubsetOf(graph.edges);
bool IsSupergraphOf(Graph graph) =>
  graph.IsSubgraphOf(this);
```

Операции с графи

```
bool IsSubgraphOf(Graph graph) =>
  vertices.IsSubsetOf(graph.vertices) &&
    edges.IsSubsetOf(graph.edges);
bool IsSupergraphOf(Graph graph) =>
  graph.IsSubgraphOf(this);
```

```
Graph total = new Graph();
total.vertices = vertices;
total.edges =
  new HashSet<GraphEdge>(
    from vertex1 in vertices
    from vertex2 in vertices
    select Edge(vertex1, vertex2));
```

```
Graph negative = Total;
negative.edges.ExceptWith(edges);
```

Множества и списъци в Java

Множества - Set<E> и HashSet<E>:

- HashSet.add, HashSet.remove, HashSet.contains
- HashSet. retainAll , HashSet.addAll, HashSet.removeAll
- HashSet. containsAll

Множества и списъци в Java

Множества - Set<E> и HashSet<E>:

- HashSet.add, HashSet.remove, HashSet.contains
- HashSet. retainAll , HashSet.addAll, HashSet.removeAll
- HashSet. containsAll

Списъци - List <E> и ArrayList <E>:

- ArrayList . add, ArrayList . remove, ArrayList . contains
- Stream.concat, Stream.sorted
- ArrayList . subList
- ArrayList . retainAll , ArrayList . addAll, ArrayList . removeAll

Множества и списъци в Java

Множества - Set<E> и HashSet<E>:

- HashSet.add, HashSet.remove, HashSet.contains
- HashSet. retainAll , HashSet.addAll, HashSet.removeAll
- HashSet. containsAll

Списъци - List <E> и ArrayList <E>:

- ArrayList .add, ArrayList .remove, ArrayList . contains
- Stream.concat, Stream.sorted
- ArrayList . subList
- ArrayList . retainAll , ArrayList .addAll, ArrayList .removeAll

Други: HashMap<K,V>, Hashtable<K,V>

Максимизираща функция

```
Map<Double, Double> function1 = ...;
Map<Double, Double> function2 = ...;

Set<Entry<Double,Double>> maximalEntries =
  function1.entrySet().stream().
    filter(entry ->
        function2.getOrDefault(entry.getKey(),
          function1.get(entry.getKey()))
        <= function1.get(entry.getKey())).
      collect(Collectors.toCollection(HashSet::new));

maximalEntries.addAll(
  function2.entrySet().stream().
    filter(entry ->
        function1.getOrDefault(entry.getKey(),
          function2.get(entry.getKey()))
        <= function2.get(entry.getKey())).
      collect(Collectors.toCollection(HashSet::new)));
```

Множества и списъци в JavaScript

Множества - Set:

- Set.add, Set. delete, Set.has
- обединение чрез .new Set ([..., ...]) , сечение и разлика чрез Set. filter
- подмножество чрез Set. filter или Set.every

Множества и списъци в JavaScript

Множества - Set:

- Set.add, Set.delete, Set.has
- обединение чрез .new Set ([..., ...]) , сечение и разлика чрез Set. filter
- подмножество чрез Set. filter или Set.every

Списъци - Array:

- Array.push, Array.pop, Array. unshift , Array. shift , Array. includes
- Array.concat, Array. slice
- Array. sort , Array. reverse

Множества и списъци в JavaScript

Множества - Set:

- Set.add, Set. delete , Set.has
- обединение чрез .new Set ([..., ...]) , сечение и разлика чрез Set. filter
- подмножество чрез Set. filter или Set.every

Списъци - Array:

- Array.push, Array.pop, Array. unshift , Array. shift , Array. includes
- Array.concat, Array. slice
- Array. sort , Array. reverse

Други: Map, Object. freeze ([ ])

Операции с множества

```
Set.prototype.Union =
  function(otherSet) {
    return new Set([...this, ...otherSet]); }

Set.prototype.Intersection =
  function(otherSet) {
    return new Set([...this].filter(
      element => otherSet.has(element))); }

Set.prototype.Difference =
  function(otherSet) {
    return new Set([...this].filter(
      element => !otherSet.has(element))); }

Set.prototype.SymmetricDifference =
  function(otherSet) {
    return this.Difference(otherSet).
      Union(otherSet.Difference(this)); }
```

Операции с множества (продължение)

```
function MultiIntersection()
{
  var minimalSizedSet =
    [...arguments].reduce(
      function(aggregate, current) {
        return
          aggregate.size <= current.size ?
            aggregate :
            current; },
      arguments[0]);

  return
    new Set([...minimalSizedSet].
      filter(element =>
        [...arguments].every(set =>
          set.has(element))));
}
```

Множества и списъци в Python

Множества - set:

- set.add, set.update, set.remove, set.pop, in, not in
- set.intersection, &, set.union, |, set.difference, −, set.symmetric_difference, ^
- set.issubset, <=, set.issuperset, >=

Множества и списъци в Python

Множества - set:

- set.add, set.update, set.remove, set.pop, in, not in
- set.intersection, &, set.union, |, set.difference, −, set.symmetric_difference, ^
- set.issubset, <=, set.issuperset, >=

Списъци - list:

- list.append, list.extend, list.insert, list.remove, list.pop
- iterable.slice, iterable.chain, iterable.zip, iterable.tee
- list.sort, iterable.sorted, list.reverse, iterable.reversed

Множества и списъци в Python

Множества - set:

- set.add, set.update, set.remove, set.pop, in, not in
- set. intersection , &, set.union, |, set. difference , −, set.symmetric_difference, ^
- set. issubset , <=, set. issuperset , >=

Списъци - list :

- list .append, list .extend, list . insert , list .remove, list .pop
- iterable . slice , iterable .chain, iterable .zip, iterable . tee
- list . sort , iterable .sorted, list . reverse , iterable . reversed

Други: dictionary , (), frozenset

Сечение на редици

```
def LazyCaterer():
  index = 0
  while True:
    yield (index * index + index + 2) / 2
    index += 1

def Catalan():
  index = 0
  current = 1
  while True:
    yield current
    current = current * (4 * index + 2) / (index + 2)
    index += 1
```

Сечение на редици

```
def LazyCaterer():
  index = 0
  while True:
    yield (index * index + index + 2) / 2
    index += 1

def Catalan():
  index = 0
  current = 1
  while True:
    yield current
    current = current * (4 * index + 2) / (index + 2)
    index += 1
```

```
def LessThan(sequence, limit):
  return takewhile(lambda n: n <= limit, sequence())

result = set(LessThan(Fibonacci, 1000000000)) &
         set(LessThan(LazyCaterer, 1000000000)) &
           set(LessThan(Catalan, 1000000000))
```

Упражнения

- Генератор на подмножества.

Упражнения

- Генератор на подмножества.
- Генератор на подсписъци.

Упражнения

- Генератор на подмножества.
- Генератор на подсписъци.
- Генератор на подредици.

Упражнения

- Генератор на подмножества.
- Генератор на подсписъци.
- Генератор на подредици.
- Генератор на пермутации.