

הדגמות אינטראקטיביות של מודלים סטטיסטיים סטוכסטיים
החוג לסטטיסטיקה, אוניברסיטת חיפה,
סמסטר חורף – תשס"ט
מרצה: יוני נצרתי

מבחן ביניים מס' 1 קבוצה ב'

29.12.2009

משך המבחן, שעתים וחצי. יש לענות על כל השאלות.
דיון פתוח בין הנבחנים – ניתן להגיש במשותף או בנפרד.

חלק א: עבור כל סעיף כתוב מהו הפלט.

(1א)

```
Fold[#1^#2&,2,Table[i,{i,3}]]
```

פתרון:

(2א)

```
Graphics[  
{Line[Table[{x,If[x<0,-x,x]},{x,-1,1,.01]}],Axes-True]
```

פתרון:

(3א)

```
Nest[D[#,x]&,x^5,5]
```

פתרון:

חלק ב: עבור כל סעיף רשום תשובה קצרה (רק את מה שיש להזין לMathematica)

(1)

כתוב פונ' המקבלת רשימה של מספרים ומחזירה את האיבר המקסימאלי ברשימה. אסור להשתמש בפונקציות Sort ו Max.

פתרון:

(2ב)

כתוב פונקציה המקבלת רשימה של זוגות $\{x,y\}$ במישור ומחזירה (מציירת) אובייקט גרפי שבו מצויר עיגול ברדיוס 0.5 סביב כל נקודה ברשימה.
הבהרה: הנה דוגמא לקלט הפונקציה: $\{\{1,2\}, \{-3,4\}, \{2,3\}\}$.

פתרון:

(3ב)

כתוב פונקציה המקבלת רשימה של מספרים. במידה והמספרים ממוינים מקטן לגדול אז הפונקציה מחזירה True. אחרת, הפונקציה מחזירה False. אסור להשתמש ב Sort.

לדוגמא:

```
IsSorted([1, 6, 8, 12]) True מחזירה  
IsSorted([1, 6, 3, 12]) False מחזירה
```

פתרון:

In[28]:= ? **Fold**

`Fold[f, x, list]` gives the last element of `FoldList[f, x, list]`. >>

In[29]:= ? **FoldList**

`FoldList[f, x, {a, b, ...}]` gives `{x, f[x, a], f[f[x, a], b], ...}`. >>

In[30]:= ? **Nest**

`Nest[f, expr, n]` gives an expression with `f` applied `n` times to `expr`. >>

In[31]:= ? **NestList**

`NestList[f, expr, n]` gives a list of the results of applying `f` to `expr` 0 through `n` times. >>

In[32]:= ? **If**

`If[condition, t, f]` gives `t` if `condition` evaluates to True, and `f` if it evaluates to False.

`If[condition, t, f, u]` gives `u` if `condition` evaluates to neither True nor False. >>

In[33]:= ? **Select**

`Select[list, crit]` picks out all elements e_i of `list` for which `crit[ei]` is True.

`Select[list, crit, n]` picks out the first `n` elements for which `crit[ei]` is True. >>

In[34]:= ? Plus

$x + y + z$ represents a sum of terms. >>

In[35]:= ? Apply

`Apply[f, expr]` or `f @@ expr` replaces the head of `expr` by `f`.

`Apply[f, expr, levelspec]` replaces heads in parts of `expr` specified by `levspec`. >>

In[36]:= ? Map

`Map[f, expr]` or `f /@ expr` applies `f` to each element on the first level in `expr`.

`Map[f, expr, levelspec]` applies `f` to parts of `expr` specified by `levspec`. >>

In[37]:= ? Table

`Table[expr, {imax}]` generates a list of i_{max} copies of `expr`.

`Table[expr, {i, imax}]` generates a list of the values of `expr` when i runs from 1 to i_{max} .

`Table[expr, {i, imin, imax}]` starts with $i = i_{min}$.

`Table[expr, {i, imin, imax, di}]` uses steps di .

`Table[expr, {i, {i1, i2, ...}}]` uses the successive values i_1, i_2, \dots

`Table[expr, {i, imin, imax}, {j, jmin, jmax}, ...]`

gives a nested list. The list associated with i is outermost. >>

In[38]:= ? OddQ

`OddQ[expr]` gives True if `expr` is an odd integer, and False otherwise. >>

In[39]:= ? Last

`Last[expr]` gives the last element in `expr`. >>

In[40]:= ? Join

`Join[list1, list2, ...]` concatenates lists or other expressions that share the same head.

`Join[list1, list2, ..., n]` joins the objects at level n in each of the `listi`. >>

In[41]:= ? ListPlot

`ListPlot[{y1, y2, ...}]` plots points corresponding

to a list of values, assumed to correspond to x coordinates 1, 2, ...

`ListPlot[{{x1, y1}, {x2, y2}, ...}]` plots a list of points with specified x and y coordinates.

`ListPlot[{list1, list2, ...}]` plots several lists of points. >>