

Optimal strategy for exam taking schedule

(scroll down for result and visualization)

```
In[2]:= Notazok = 50; (* Examination questions pool *)
Nterminov = 3; (*How many exam sessions*)
NpytaSa = 4; (*Amount examined*)

In[3]:= pSpravimSkusku[paramPocetOtazok_, paramPocetViem_, paramPytaSa_] :=
  1 - (CDF[HypergeometricDistribution[paramPytaSa,
    paramPocetViem, paramPocetOtazok]]) [paramPytaSa / 2 - 1]
```

Study (nearly) everything for the last exam

```
In[1]:= kolkoChcemVediet = Notazok - (NpytaSa / 2 // Floor);

In[4]:= mozneRozdeleniaNaTerminy = Flatten[Table[{i, j, kolkoChcemVediet},
  {i, 1, kolkoChcemVediet, 1}, {j, i, kolkoChcemVediet, 1}], 1];
```

Calculate probabilities of passing on n-th session, based on study effort distribution

```
In[5]:= Clear[pravdepodobnostNaNtyTermin];
pravdepodobnostNaNtyTermin[paramPocetOtazok_,
  paramPytaSa_, paramRozdelenie_List] := (pPrvyPokus =
  pSpravimSkusku[paramPocetOtazok, paramRozdelenie[[1]], paramPytaSa];
  pDruhyPokus = (1 - pPrvyPokus) * pSpravimSkusku[
    paramPocetOtazok, paramRozdelenie[[2]], paramPytaSa];
  pTretiPokus = (1 - pPrvyPokus) *
    (1 - pSpravimSkusku[paramPocetOtazok, paramRozdelenie[[2]], paramPytaSa]) *
    pSpravimSkusku[paramPocetOtazok, paramRozdelenie[[3]], paramPytaSa];
  {pPrvyPokus, pDruhyPokus, pTretiPokus})

(*Test if correct*) {Total[#[[1]], #[[2]]] & /@
  ({pravdepodobnostNaNtyTermin[Notazok, NpytaSa, #], #} & /@
  mozneRozdeleniaNaTerminy) // N;
```

```
In[14]:= Clear[vynalozenaPraca];
vynalozenaPraca[paramPocetOtazok_, paramPytaSa_, paramRozdelenie_List] :=
  (trojicaPrav = pravdepodobnostNaNtyTermin[
    paramPocetOtazok, paramPytaSa, paramRozdelenie];
  trojicaUcenie = {paramRozdelenie[[1]], paramRozdelenie[[2]] -
    paramRozdelenie[[1]], paramRozdelenie[[3]] - paramRozdelenie[[2]]};
  Return[paramRozdelenie * trojicaPrav]
  (* "S takou pravdepodobnostou sa este doucit treba _ " *)
  )

(*Verify → ak vynalozenaPraca vracia "trojicaUcenie" *)
Total /@ (vynalozenaPraca[Notazok, NpytaSa, #] & /@ mozneRozdeleniaNaTerminy) ;
```

Explore expected effort based on concrete examples of effort distribution

```
In[15]:= vynalozenaPraca[Notazok, NpytaSa, {28, 28, 28}] // Total // N
```

```
Out[15]= 27.7059
```

```
In[16]= vynalozenaPraca[Notazok, NpytaSa, {0, 0, 28}] // Total // N
```

```
Out[16]= 21.8681
```

```
In[17]= vynalozenaPraca[Notazok, NpytaSa, {15, 15, 28}] // Total // N
```

```
Out[17]= 17.9344
```

```
In[18]= pracaDB = {vynalozenaPraca[Notazok, NpytaSa, #], #} & /@mozneRozdeleniaNaTerminy;
```

```
In[19]= pracaArozdenie = {Total#[#[[1]]] // N, #[[2]]} & /@pracaDB;
```

```
In[20]= najmesiaOcakavanaPraca = pracaArozdenie[[All, 1]] // Min;
```

Lowest expected effort: distribute learning to 19/25/48 studied questions for 1st/2nd/3rd session

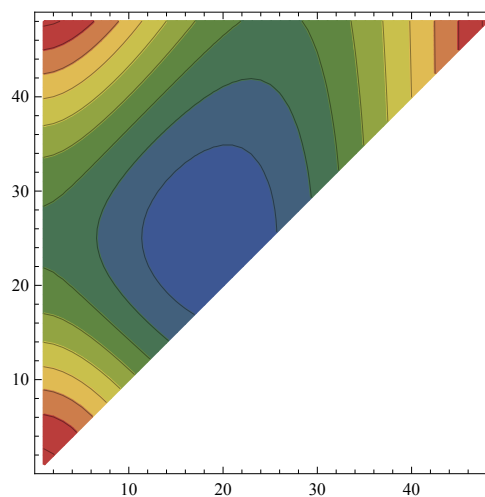
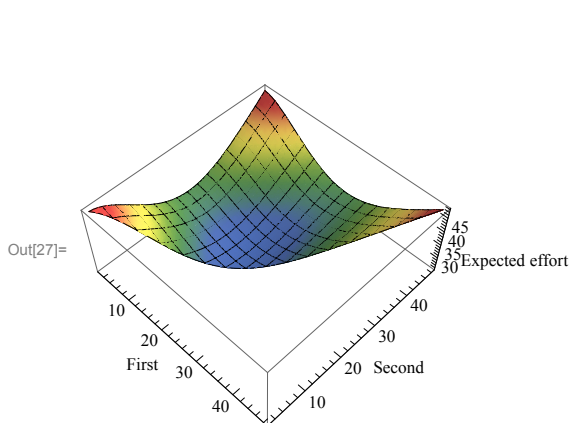
```
Cases[pracaArozdenie, {najmesiaOcakavanaPraca, _}]
```

```
{{25.6001, {19, 25, 48}}}
```

Visualization of different distributions

```
In[27]= GraphicsRow@
```

```
{ListPlot3D[{#[[1]], #[[2]], vynalozenaPraca[Notazok, NpytaSa, #] // Total} & /@
  mozneRozdeleniaNaTerminy, ColorFunction -> "DarkRainbow",
  AxesLabel -> {"First", "Second", "Expected effort"}], ListContourPlot[
  {#[[1]], #[[2]], vynalozenaPraca[Notazok, NpytaSa, #] // Total} & /@
  mozneRozdeleniaNaTerminy,
  ColorFunction -> "DarkRainbow", AxesLabel -> {"First", "Second"}]}
```



One-session only:

