TÖL403G GREINING REIKNIRITA

5. Endurrakning 1

Hjálmtýr Hafsteinsson

Vor 2022

Í þessum fyrirlestri

- Endurrakning (*backtracking*)

- *N* drottningar (*n queens*)
- Leikjatré (*game trees*)
- Hlutmengissumma (*subset sum*)
- Strengskipting (*text segmentation*)

2.1 − 2.5

Endurrakning (*backtracking*)

- Endurkvæm lausnaraðferð

- Brýtur verkefnið upp í smærri hlutverkefni
- Ef velja þarf á milli leiða;
  - Fara **allar leiðir** endurkvæmt
  - Velja svo þá með bestu útkomuna

"Ofbeldisaðferð"
(*brute force*)

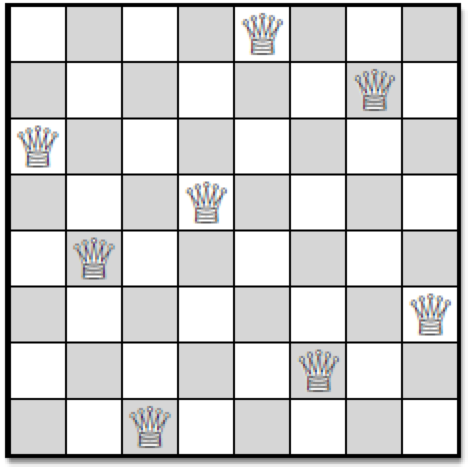
- Fáum oftast veldistíma reiknirit með endurrakningu

- Sjáum næst kvika bestun (*dynamic programming*)
  - Getum þá oft endurbætt þessar aðferðir með því að geyma milliniðurstöður

N drottningar

- Upphaflega sett fram fyrir 8x8 skákborð
- Setja 8 drottingar á skákborð þannig að engin þeirra ógni annari
  - þ.e. engar tvær drottingar á sömu línu, dálki eða skálínu
- Almennt: Staðsetja *n* drottningar á *nxn* skákborði

- Mikill fjöldi möguleika sem þarf að athuga
  - Fyrir 8x8 verkefnið eru $C(64,8) = 4.426.165.368$ möguleikar
  - Af þeim eru 92 löglegar lausnir
  - Fjöldi möguleika vex mjög hratt með hækkandi gildi á *n*
    - 10x10 verkefnið hefur C(100, 10) ~$1.7*10^{13}$, löglegar lausnir 724
    - 20x20 verkefnið hefur C(400, 20) ~$2.8*10^{33}$, löglegar lausnir ~3 milljarðar



Ein möguleg lausn á 8x8 verkefninu

Endurrakningarreiknirit

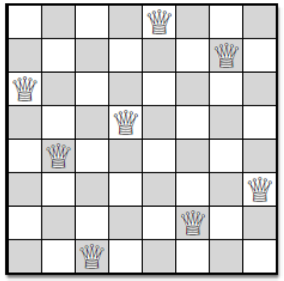
Hvað eru mörg 'R' í því?

```
PLACEQUEENS(Q[1..n], r):
  if r = n + 1
      print Q[1..n]
  else
      for j ← 1 to n
          legal ← TRUE
          for i ← 1 to r − 1
              if (Q[i] = j) or (Q[i] = j + r − i) or (Q[i] = j − r + i)
                  legal ← FALSE
          if legal
              Q[r] ← j
              PLACEQUEENS(Q[1..n], r + 1)      ⟨⟨Recursion!⟩⟩
```

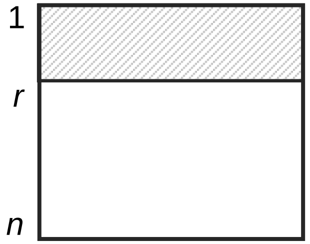
Notum fylkið $Q[1..n]$ fyrir staðsetningu drottninganna

$Q[i]$ er dálkanúmer drottningar í línu i

Í lausninni á síðustu glæru væri
Q = [5, 7, 1, 4, 2, 8, 6, 3]

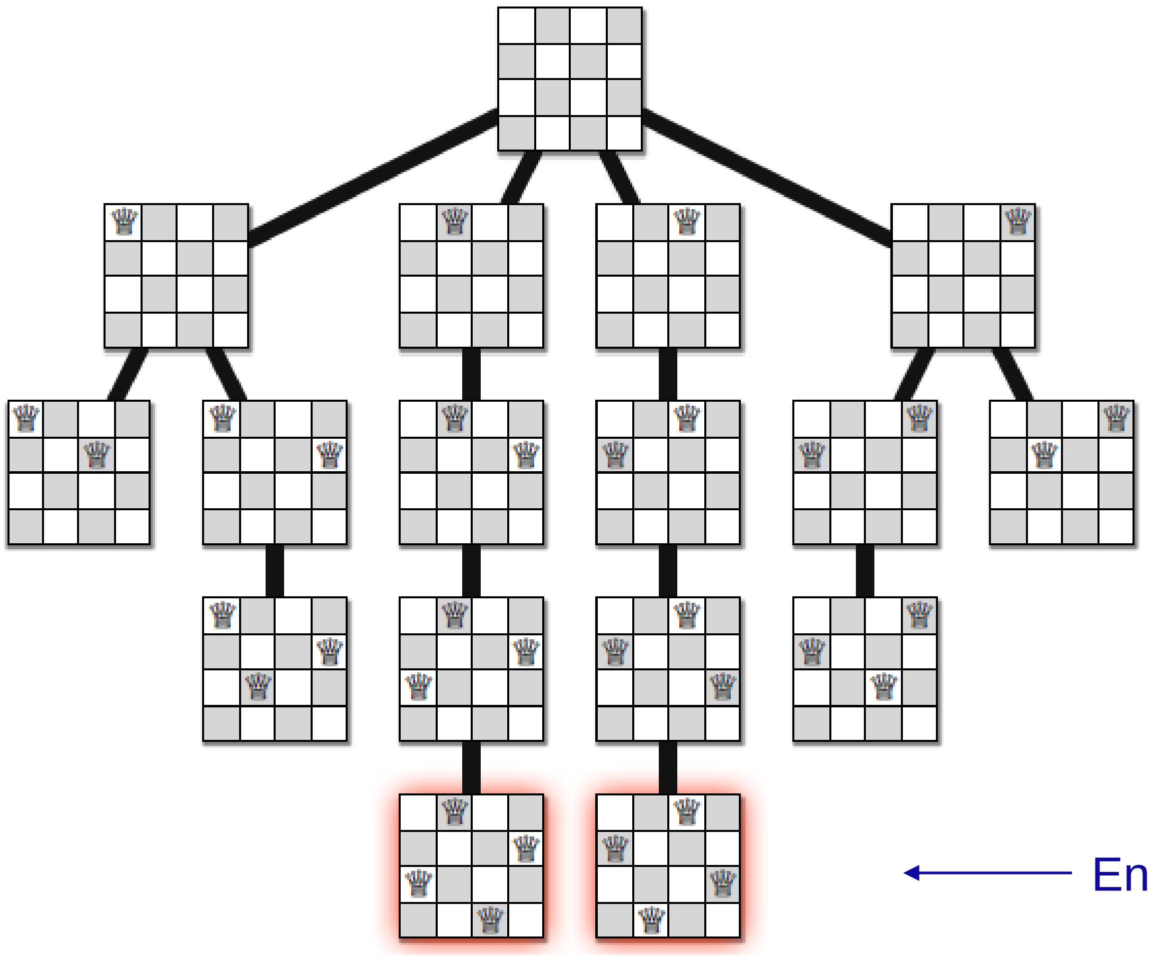


Í upphafi fallsins er viðfangið r númer fyrstu auðu línunnar og $Q[1..r-1]$ inniheldur staðsetningar fyrstu r-1 drottninganna



Fyrir hvern mögulegan dálk (ytri lykkja) er athugað í innri lykkju hvort sá reitur ógni einhverri drottingu sem þegar er komin í línur 1 til r-1

Hermun fyrir 4x4 verkefni



4 möguleikar á fyrstu línu

Fyrir hvern þeirra eru 1
eða 2 löglegir möguleikar

Stundum er ekki hægt að gera neitt.
Losnum þá við að rekja niður þessa grein

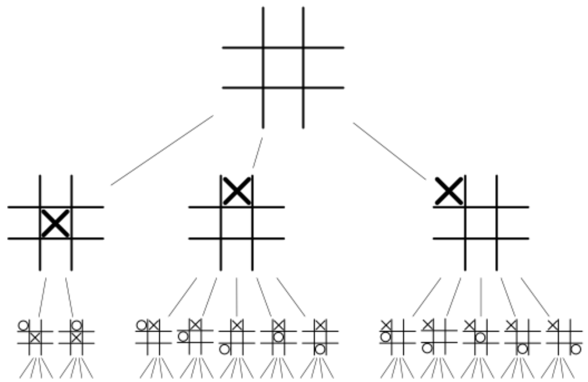
Endum með tvær lausnir

Heildarfjöldi möguleika er
C(16, 4) = 1820

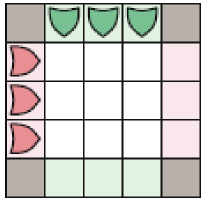
Þurfum ekki að skoða nema
hluta af þeim

Leikjatré (*game trees*)

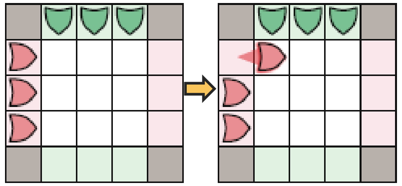
- Notuð til að halda utanum mögulegar stöður í tveggja manna leikjum
  - Til dæmis skák, go, mylla (*tic-tac-toe*), ...
- Hver <u>hnútur</u> í trénu er <u>staða leiksins</u> á tilteknum tíma
  - Ásamt því hvor leikmaðurinn á leik
- Það er <u>ör</u> frá hnúti *x* til hnútar *y* ef til er löglegur leikur sem <u>breytir stöðu *x* yfir í stöðu *y*</u>
- Rót trésins er upphafsstaða leiksins
- Sérhver vegur frá rót til laufs er löglegur kláraður leikur
- Leikjatré geta verið mjög stór
  - Fullkomið leikjatré fyrir myllu hefur hæð 9 og 255.168 hnúta



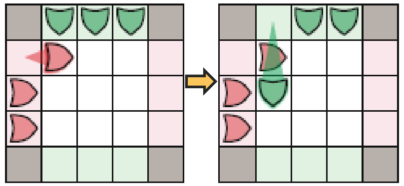
Sykurmolaleikurinn

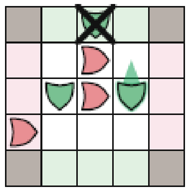


- Skoðum tveggja manna leik sem spilaður er með "sykurmolum" á 3x3 grind
- *Rauður* byrjar með sína mola á vinstri hliðinni og *Grænn* á efri hlið
- Leikmenn mega færa einn mola um eitt skref í einu
  - Rauður alltaf til hægri og Grænn alltaf niður
- Ef næsti reitur er auður þá færist molinn um einn reit

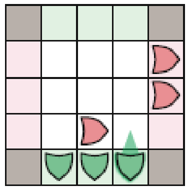


- Ef næsti reitur er ekki auður þá hoppar molinn yfir
  - Það má þó aðeins hoppa yfir einn reit



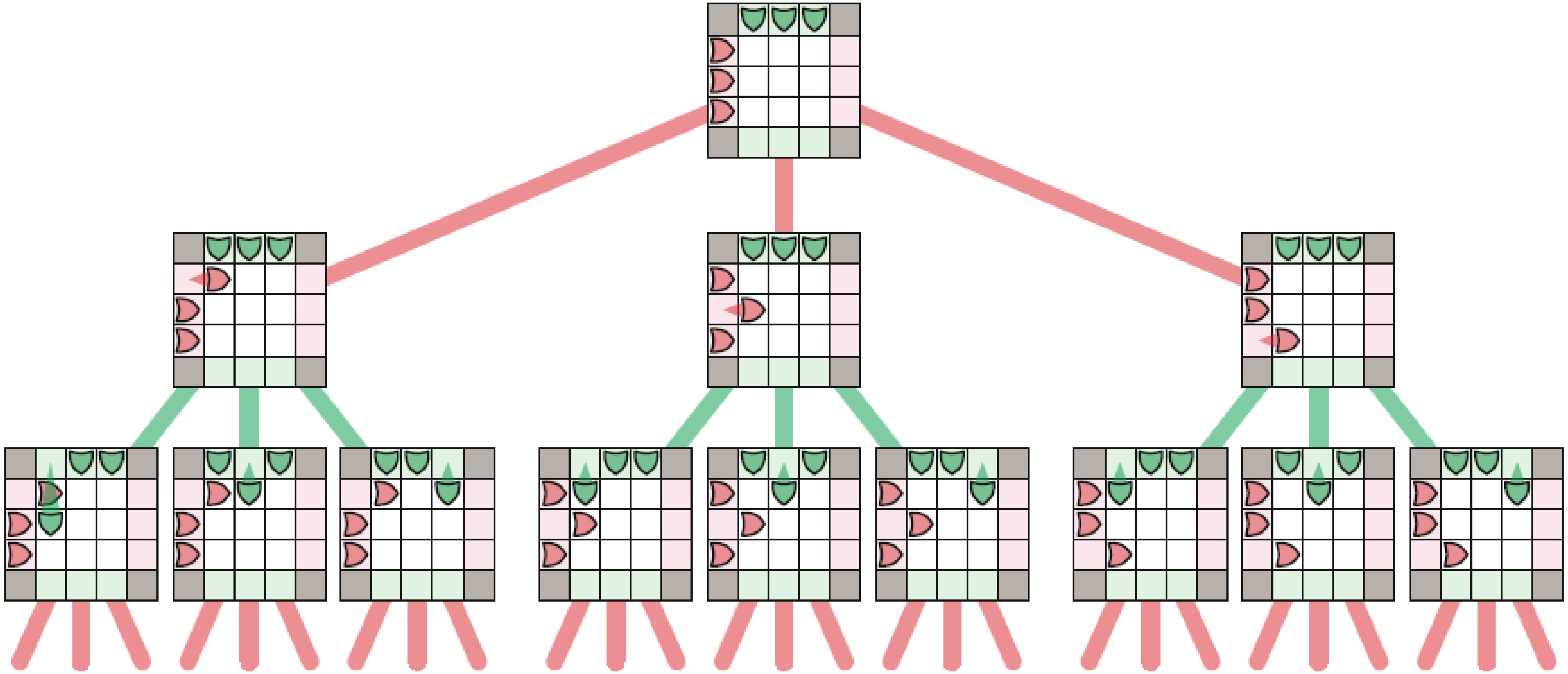


- Sá leikmaður sem fyrstur nær að koma öllum sínum molum yfir á hina hlið grindarinnar vinnur leikinn



Lokastaða,
Grænn vann

Leikjatré fyrir sykurmolaleik



Fyrstu tvö lög leikjatrésins. Einn leikur fyrir hvorn leikmann

Góðar og slæmar stöður

- Leikjastaða (*game state*) er **góð** ef núverandi leikmaður hefur unnið leikinn, eða ef núverandi leikmaður getur fært yfir í **slæma** stöðu fyrir andstæðinginn
- Leikjastaða er **slæm** ef núverandi leikmaður hefur tapað leiknum, eða ef sérhver mögulegur leikur færir yfir í **góða** stöðu fyrir andstæðinginn

- Þá gildir:
  - Innri hnútur (ekki lauf) er **góður** ef hann hefur a.m.k. eitt **slæmt** barn
  - Ef leikmaður er í **góðri** stöðu þá getur hann unnið leikinn, jafnvel þó andstæðingurinn leiki fullkomlega
  - Ef leikmaður er í **slæmri** stöðu þá getur hann aðeins unnið ef andstæðingurinn gerir mistök

Finna gæði stöðu með endurrakningu

Djúpleit í gegnum allt leikjatréð − reiknar gæði staðanna á leið upp endurkvæmnina

$\underline{\textsc{PlayAnyGame}(X, \mathit{player}):}$
    if *player* has already won in state X
        return Good
    if *player* has already lost in state X
        return Bad
    for all legal moves $X \rightsquigarrow Y$
        if $\textsc{PlayAnyGame}(Y, \neg\mathit{player}) = \textsc{Bad}$
            return Good   ⟨⟨$X \rightsquigarrow Y$ is a good move⟩⟩
    return Bad   ⟨⟨There are no good moves⟩⟩

Stoppskilyrði

Ef einhver leið til að færa í slæma stöðu fyrir andstæðing þá erum við í góðri stöðu!

Ef engin góð staða fannst þá erum við í slæmri stöðu!

Hægt að skera burt (*prune*) greinar trésins sem ekki geta leitt til góðrar niðurstöðu

Þetta er mikilvægt fyrir leiki með stór leikjatré, eins og skák

Æfing um leikjatré

- Skoðið reikniritið *PlayAnyGame* á síðustu glæru. Skiptir máli í hvaða röð farið er niður greinar trésins?
  - Hvaða áhrif hefur það?
  - Hver ætti röðin að vera (ef það skiptir máli)?

Hlutmengissumma (*subset sum*)

- Höfum mengi *X* af jákvæðum heiltölum og heiltöluna *T*
- Er til hlutmengi í *X* með summuna *T*?
- Dæmi:

Athugið að röð talnanna skiptir ekki máli (*X* er mengi)

$X = \{8, 6, 7, 5, 3, 10, 9\}$ og $T = 15$

Þá er til hlutmengið {8, 7} með summuna 15, sömuleiðis {5, 10} og {6, 9} og svarið er **satt**

en ef X = {11, 6, 5, 1, 7, 13, 12} og T = 15?

Þá er svarið **ósatt**

Manni sýnist að það þurfi að prófa alla möguleika

Endurkvæm endurrakning

Er til hlutmengi í X með summuna 0?

Jaðartilvik:

Ef $T = 0$ þá skila **satt**, því $\varnothing$ virkar

Ef $T < 0$ eða ef $T > 0$ en X er tómt þá skila **ósatt**

Er til hlutmengi í $\varnothing$ með summuna 5?

Almenn tilvik: Skoðum stak x í X. Annað hvort er x í hlutmengi með summu T eða x er ekki í hlutmengi með summu T

Ef x er í hlutmengi með summu T þá
er til hlutmengi í $X \setminus \{x\}$ með summu $T - x$

Annars
er til hlutmengi í $X \setminus \{x\}$ með summu T

Reiknirit fyrir hlutmengissummu

```
⟨⟨Does any subset of X[1..i] sum to T?⟩⟩
SubsetSum(X, i, T):
  if T = 0
      return True
  else if T < 0 or i = 0
      return False
  else
      with ← SubsetSum(X, i − 1, T − X[i])   ⟨⟨Recurse!⟩⟩
      wout ← SubsetSum(X, i − 1, T)          ⟨⟨Recurse!⟩⟩
      return (with ∨ wout)
```

Köllum á fallið í upphafi með *SubsetSum*(*X*, *n*, *T*)

Mengið er geymt í fylki

Mengið "minnkar" um 1 í hverju endurkvæma kalli

Tími: $T(n) = 2T(n-1) + O(1)$

Lausn: $T(n) = O(2^n)$

Getum notað svipað reiknirit til að <u>finna</u> hlutmengi sem passar

Sömuleiðis hægt að finna <u>fjölda</u> hlutmengja sem passa

Æfing fyrir hlutmengissummu

- Við höfum X = [6, 3, 5, 9, 7], T = 10 og i = 3.
- Hvað gerist í fallinu *SubsetSum*(X, i, T) fyrir þessi gildi?

```
⟨⟨Does any subset of X[1..i] sum to T?⟩⟩
SubsetSum(X, i, T):
  if T = 0
      return True
  else if T < 0 or i = 0
      return False
  else
      with ← SubsetSum(X, i − 1, T − X[i])    ⟨⟨Recurse!⟩⟩
      wout ← SubsetSum(X, i − 1, T)           ⟨⟨Recurse!⟩⟩
      return (with ∨ wout)
```

- En ef T = 8?

Strengskipting (*text separation*)

- Höfum samfelldan textastreng, viljum brjóta hann upp í orð
- Dæmi:

BOTHEARTHANDSATURNSPIN

getum skipt upp sem BOTH-EARTH-AND-SATURN-SPIN

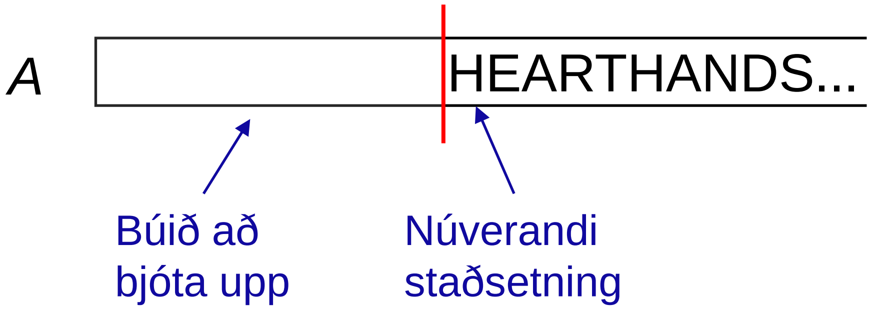
eða sem BOT-HEART-HANDS-AT-URNS-PIN

Allt lögleg ensk orð

Höfum fallið **IsWord(*w*)**, sem skilar **<u>true</u>** ef *w* er löglegt enskt orð, en **<u>false</u>** annars

Endurrakning fyrir strengskiptingu

Prófum alla möguleika og hugsum ekki um það sem kemur á undan



Endurkvæmt kall

Er "H" orð? **NEI!**

Er "HE" orð? **JÁ!** Er restin skiptanleg?

Er "HEA" orð? **NEI!**

Er "HEAR" orð? **JÁ!** Er restin skiptanleg?

....

Prófum alla möguleika

Reikniritfyrir orðskiptingu

Fáum endurkvæma reikniritið:

```
SPLITTABLE(A[1..n]):
  if n = 0
      return TRUE
  for i ← 1 to n
      if ISWORD(A[1..i])
          if SPLITTABLE(A[i+1..n])
              return TRUE
  return FALSE
```

Grunntilvik:
tómi strengurinn

Er restin skiptanleg?

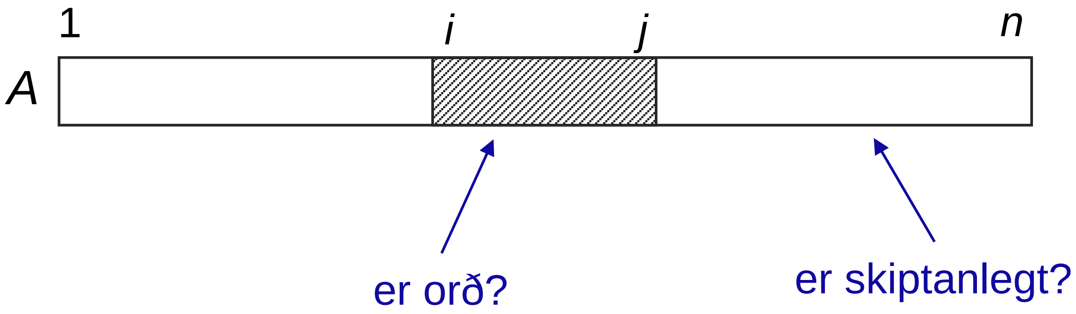
Reyndar óþarfi að senda fylkið *A* sem viðfang í fallið

Látum *A* vera víðvært og notum bara vísinn (*index*) *i*

Vísaútgáfa af orðskiptingu

Skilgreinum *Splittable*(i) = **satt** þþaa eftirstrengurinn (*suffix*) $A[i..n]$ er skiptanlegur í orð

$$Splittable(i) = \begin{cases} \underline{\textbf{satt}} & \text{ef } i > n \\ \bigvee_{j=1}^{n} IsWord(i,j) \wedge Splittable(j+1) & \text{annars} \end{cases}$$




```
⟨⟨Is the suffix A[i..n] Splittable?⟩⟩
SPLITTABLE(i):
  if i > n
      return TRUE
  for j ← i to n
      if ISWORD(i, j)
          if SPLITTABLE(j + 1)
              return TRUE
  return FALSE
```

Reiknirit fyrir víðvært fylki *A*

Tími: $O(2^n)$

Afbrigði af strengskiptingu

- Oft eru til margar skiptingar á sama strengnum
- Viljum þá kannski fá bestu skiptinguna út frá einhverju stigafalli
  - Til dæmis gefa mörg stig fyrir löng orð, eða algeng orð, ...

Höfum fallið *Score*(*w*), sem skilar stigagildi orðsins *w* (0 ef ekki löglegt orð)

Skilgreinum *MaxScore*(*i*) = hámarksstig orðskiptingar á *A*[*i*..*n*]

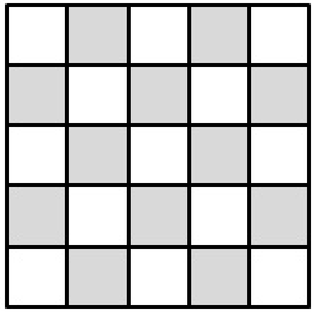
$$MaxScore(i) = \begin{cases} 0 & \text{ef } i > n \\ \max\limits_{i \le j \le n} \left(Score(A[i..j]) + MaxScore(j+1) \right) & \text{annars} \end{cases}$$

Getum notað nánast sama reikniritið til að leysa þetta verkefni

... með tímaflækjuna $O(2^n)$

Fyrirlestraæfingar

1. Finnið eina lausn fyrir 5 drottninga verkefnið:



2. Gefið mengið $X = \{3, 4, 7, 8, 20, 21, 34, 41\}$. Er til hlutmengi í því með summuna 50?

3. Hvaða ensk orð koma til greina sem fyrsta orðið í strengnum "WEEKLYASSIGNMENT"?