1) Gegeben 2 Binärwörter u, v. Repräsentieren 2 Knoten im Hypercube (Dimension k, Wort Länge k)

Routing von u nach v.

Ziel: flippen aller bits in u, die sich von v unterscheiden (transition entlang von Kanten).

- Invariante: vor dem i-ten Schleifendurchlauf wurden die ersten $i-1$ bits in u jeweils geflippt oder nicht, sodass sie mit den ersten $i-1$ bits in v übereinstimmen.
- Initialisierung: Im ersten Durchlauf wird das 1. bit in u geflippt/nicht geflippt sodass $u_1 = v_1$ ($\hat{=}$ Kantentransition)
- Erhaltung: In jedem Durchlauf i wird u_i geflippt wenn $u_i \neq v_i$ (u an v anpassen, Kantentransition)
- Terminierung: Im letzten Durchlauf wird u_k geflippt falls $u_k \neq v_k$. ~~Alle~~ Für alle u_i mit $i < k$ gilt schon $u_i = v_i$.

Nach dem letzten Durchlauf gilt $\forall_i\ 1 \leq i \leq k: u_i = v_i$

$\Rightarrow$ Erfolgreiches Routing von u nach v.

Laufzeit:

k Schleifendurchläufe für Hypercube der Dimension k (Binärwörter der Länge k). k bits müssen überprüft werden $\Rightarrow$ Laufzeit: $O(k)$.

2)

Es gibt $k!$ Wege der Länge k in einem Hypercube der Dimension k (von $(0, \dots 0)$ bis nach $(1, \dots, 1)$).

- Begründung:
  - Um von $(0, \dots, 0)$ entlang von k Kanten nach $(1, \dots, 1)$ zu gelangen, ist es notwendig alle bits zu flippen.
  - Jeder Knoten ist über k Kanten mit k Knoten verbunden. ~~die~~ Beim Verfolgen jeder Kante wird jeweils eines der k bits geflippt.
  - Wir wollen jetzt alle bits flippen. Beim Knoten $(0, \dots, 0)$ kann ein beliebiges bit geflippt werden (egal welches wir flippen, wir bewegen uns richtung $(1, \dots, 1)$. Beim nächsten Knoten können nur mehr $k-1$ bits geflippt werden, sonst wird ein 1er geflippt & wir bewegen uns Richtung $(0, \dots, 0)$. In jedem darauffolgenden Knoten kann jetzt um 1 bit weniger geflippt werden. Am vorletzten Knoten schließlich nur mehr 1 bit (alles andere bedeutet eine Bewegung richtung $(0, \dots, 0)$

$\Rightarrow$ mögliche Wege: $k \cdot (k-1) \cdot (k-2) \cdot \dots \cdot 1 = k!$