

Úvodné informácie k predmetu Informatika Algoritmy a algoritmizácia úloh

doc.Ing. Marcela Hallová, PhD.

Obsahová náplň predmetu

- Algoritmizácia a vývojové diagramy
- Hardvér a softvér počítačov
- Počítačové siete
- Mobilné siete
- Bezpečnosť na sieti – infiltrácie
- Informačná bezpečnosť
- Softvérové riadenie projektov
- MS WORD
- MS EXCEL

Podmienky absolvovania predmetu

Podmienky na udelenie zápočtu:

Na zápočet je potrebné dosiahnuť aspoň **45 bodov!**

Skúška - teoretický test z ostatných okruhov tém prednášok (okrem Algoritmizácie a VD) - **30 bodov**

Úspešné absolvovanie 2 testov:

1. test - Algoritmizácia a VD - **30 bodov**

2. test - MS Word + MS Excel - **40 bodov**



Študijné materiály a kontakty

- **LMS Moodle** – cez stránku Fakulty ekonomiky a manažmentu – fem.uniag.sk
- **Kontakt:** marcela.hallova@uniag.sk
Fakulta ekonomiky a manažmentu – Ústav účtovníctva a informatiky – 2. poschodie, č.d. 211

Algoritmus a algoritmizácia úloh

- **Prečo algoritmy?**

- Možnosť realizovať danú postupnosť krokov iným človekom, resp. strojom.
- Postup vyjadrený pomocou presne definovaných pravidiel a postupov sa stáva zrozumiteľný nielen pre autora, ale aj ďalších realizátorov.

Čo je algoritmus?

Presný a jednoznačný predpis postupu riešenia úlohy.

Proces transformácie vstupných údajov na výsledok.

Postupnosť elementárnych krokov (operácií), ktoré sa vykonávajú v určitom poradí.

Pôvod slova algoritmus

- **9. storočie** – Muhammad ibn Músá Al-Chwárizmí – geometrické riešenie kvadratických rovníc – vymyslel jednoduchý algoritmus pre násobenie dvojciferného čísla jednociferným.
- **Algoritmický spôsob riešenia úloh** – Egypt – Moskovský papyrus (asi 1850 p.n.l.) – postup výpočtu objemu zrezanej pyramídy

Čo môže byť algoritmom

návod na zostavenie nábytku z jednotlivých dielov,

postup riešenia matematickej úlohy,

recept na prípravu jedla,

program zapísaný v ľubovoľnom programovacom jazyku...

Vlastnosti algoritmov

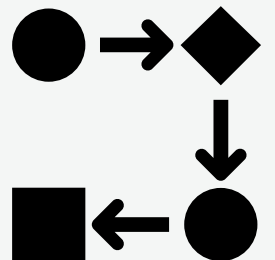
Determinovanosť

Rezultatívnosť

Hromadnosť

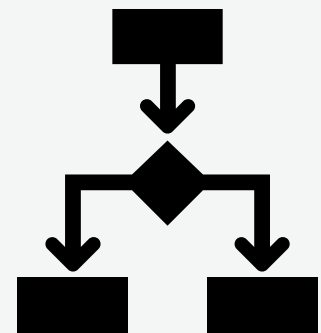
Determinovanosť

- algoritmus determinuje, t.j. presne určuje proces pretvárania vstupných údajov na výsledky,
- v každom kroku musí byť presne určené, ktorý krok sa vykoná ako ďalší,
- uskutočňovanie operácií nezávisí od vykonávateľa algoritmu.



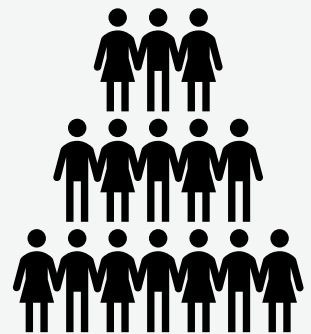
Rezultatívnosť

- pre ľubovoľnú n -ticu vstupných údajov z určitej množiny M vedie algoritmus vždy po konečnom počte krokov k hľadanému výsledku,
- množinu M nazývame "oblasť použiteľnosti daného algoritmu".



Hromadnosť

- algoritmus musí byť zostavený tak, aby riešil veľkú, obyčajne nekonečnú triedu úloh rovnakého typu,
- musí to byť popis riešenia nie jednej konkrétnej úlohy, ale celej skupiny príbuzných úloh, ktoré sa odlišujú len hodnotami vstupných údajov.



Ďalšie vlastnosti algoritmov

- Okrem základných vlastností (determinovanosť, hromadnosť a rezultatívnosť) by algoritmus mal byť:
 - **elementárny,**
 - **efektívny,**
 - **čitateľný,**
 - **modifikovateľný.**

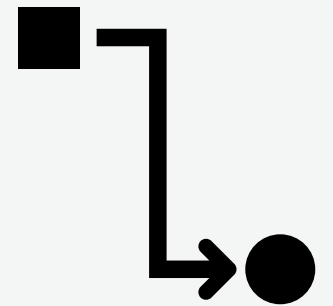


Elementárnosť algoritmu

- každý postup je možné realizovať rôznymi spôsobmi,
- pri návrhu algoritmu je potrebné dbať na to, aby jednotlivé kroky boli pre vykonávateľa algoritmu zrozumiteľné a jednoznačné.

Efektívnosť algoritmu

- určuje sa najmä vzhľadom na potrebný výpočtový čas a požadovanú kapacitu pamäti,
- efektívny algoritmus je taká postupnosť krokov, ktorá daný problém rieši s minimálnym počtom použitých prostriedkov v čo najkratšom čase.



Čitateľnosť a modifikovateľnosť algoritmu

- je vhodné navrhovať algoritmus, ktorý je vhodne rozdelený na menšie, relatívne samostatné, logicky na seba nadväzujúce celky,
- štruktúrovaný algoritmus je ľahšie pochopiteľný a modifikovateľný.

Etapy algoritmizácie úloh

Formulácia úlohy.

- jasná a jednoznačná formulácia a identifikácia
- stanovenie cieľa riešenia úlohy

Analýza úlohy.

- nájdenie algoritmu riešenia

Zostavenie riešiaceho algoritmu.

- syntetická etapa - popísanie logiky a postupu riešenia
- výsledkom je riešiaci algoritmus

Algoritmizácia

Schopnosť aktívne vytvárať algoritmy určené pre nemysliace zariadenia.

Je nevyhnutnou súčasťou schopnosti programovať na počítači.

Na tvorbu efektívnych a správnych algoritmov nestačí len zvládnutie algoritmického či programovacieho jazyka.

Je potrebná znalosť problémového prostredia a skúsenosť s formulovaním algoritmov.

Programovanie

- Program je algoritmus napísaný v programovacom jazyku
- Programovanie je konštruktívna myšlienková, ale aj praktická činnosť, kedy vytvárame nové programové produkty realizovateľné na počítači.
- Programovaním sa učíme myslieť, organizovať svoje myšlienky a dokázať ich realizáciou poveriť počítač...

Vytvorenie programu - činnosti

Algoritmizácia daného problému -
určenie vstupných a výstupných
podmienok.



Vytvorenie programu (programového
produktu) a vhodnej programovej
dokumentácie.



Zapísanie a odladenie programu
priamo na počítači.

Algoritmický jazyk

- Na zápis algoritmov bolo potrebné vytvoriť formálne jazyky - umelo vytvorené špeciálne na tento účel, tzv. algoritmické jazyky.
- Rozlišujeme dve skupiny:
 - **Grafické algoritmické jazyky** - vývojové diagramy, štruktúrogramy...
 - **Lineárne algoritmické jazyky** - napr. slovný zápis v národnom jazyku, programovací jazyk...



Algoritmický jazyk by mal svojou konštrukciou napomáhať splneniu vlastností algoritmu.



V algoritmických jazykoch sú 2 zvýraznené zložky:

operačná zložka

riadiaca zložka

Operačná zložka

Elementárne činnosti, ktoré dokáže procesor vykonávať.

Základnými činnosťami sú príkazy a podmienky.

Príkazy - vety jazyka, ktoré „prikazujú“ procesoru vykonať isté, presne stanovené činnosti. Napr. príkazy vstupu a výstupu, príkaz priradenia...

Príkazy musia spracovávať nejaké objekty - v programovaní sú to premenné, konštanty a výrazy.

Operačná zložka

Premenná - objekt, ktorý obsahuje počas realizácie algoritmu konkrétnu hodnotu presne stanoveného typu (celé číslo, reálne číslo...)

Konštanta - objekt, ktorý počas celej realizácie algoritmu nadobúda jedinú konkrétnu hodnotu príslušného typu (napr. π)

Výraz - predpis, ktorý obsahuje konštanty, premenné a spôsoby ich spracovania pomocou operácií a funkcií. Jeho výsledkom je hodnota príslušného typu.

Riadiaca zložka

V algoritmickom jazyku musí byť presne stanovené poradie vykonávania jednotlivých činností, aby realizátor (procesor) nemusel uvažovať, čo má kedy vykonať.

Musí byť určená presná postupnosť krokov.

Zápis algoritmu

- Algoritmom môže byť napr. výpočtový proces, popis geometrickej konštrukcie, právny predpis, návod na vykonanie určitej činnosti a pod., je možné ho vyjadriť rôznymi prostriedkami:
 - slovným popisom,
 - matematickými symbolmi (rovnice, matice, vzorce a pod.),
 - konštrukčným postupom,
 - grafickými značkami.

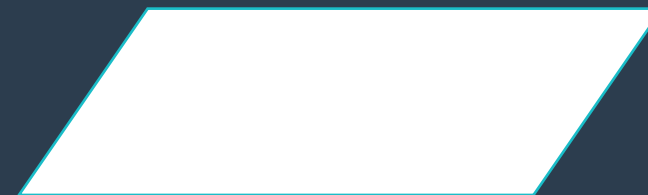
Vývojový diagram

- grafická forma zápisu algoritmu,
- blokový diagram určitého procesu, ktorý vyjadruje jeho štruktúru a nadväznosť jednotlivých operácií,
- na kreslenie vývojových diagramov používame grafické značky.



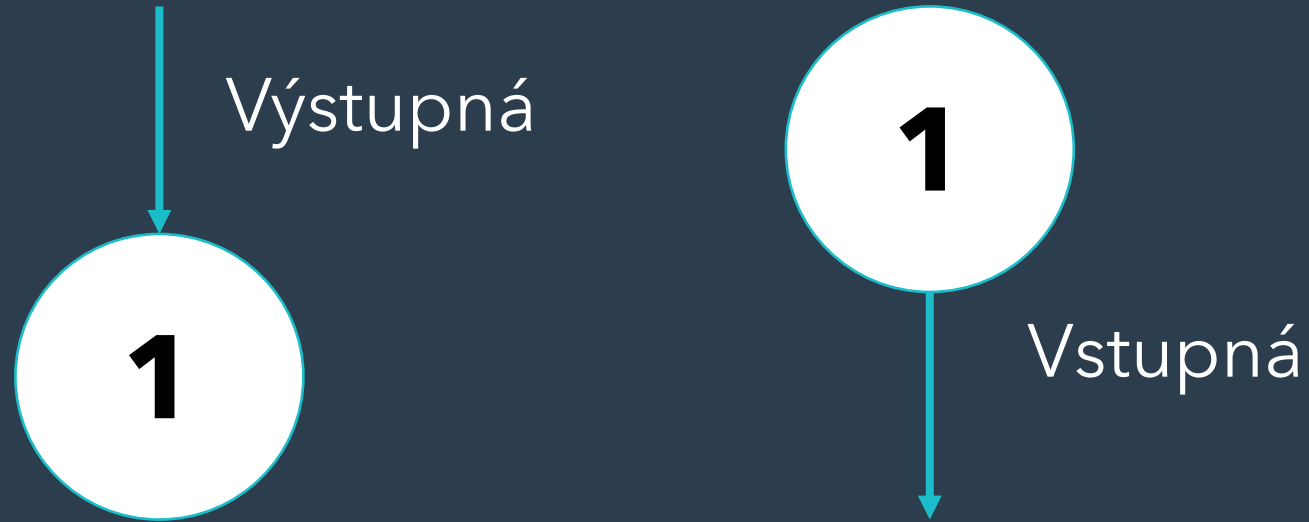
Základné značky VD

- značka pre spracovanie
- značka pre vetvenie, rozhodovanie
- hraničná značka - začiatok, koniec
- značka pre vstup/výstup



Základné značky VD

- spojky



- spojnice



Druhy vývojových diagramov

Hrubé VD

- riešenie rozsiahlych úloh, ktoré sú rozdelené na niekoľko čiastkových úloh (podprogramov)

Analytické VD

- podrobnejšie algoritmy riešenia

VD programu

- detailne rozpracovaný postup riešenia – v každom bloku VD je spravidla jedna operácia

Vývojové diagramy programu

VD priame - nedochádza k žiadnemu alternatívnemu postupu, proces je priamočiary (napr. sčítanie 4 čísel).

VD s vetvením:

Bez cyklu - bez opakovania (napr. maximum z 3 čísel)

S cyklom - s opakovaním (napr. súčet prvkov vektora, matice)

Pochopenie pojmu algoritmus a procesu algoritmizácie si môžeme vysvetliť na nasledujúcom príklade:

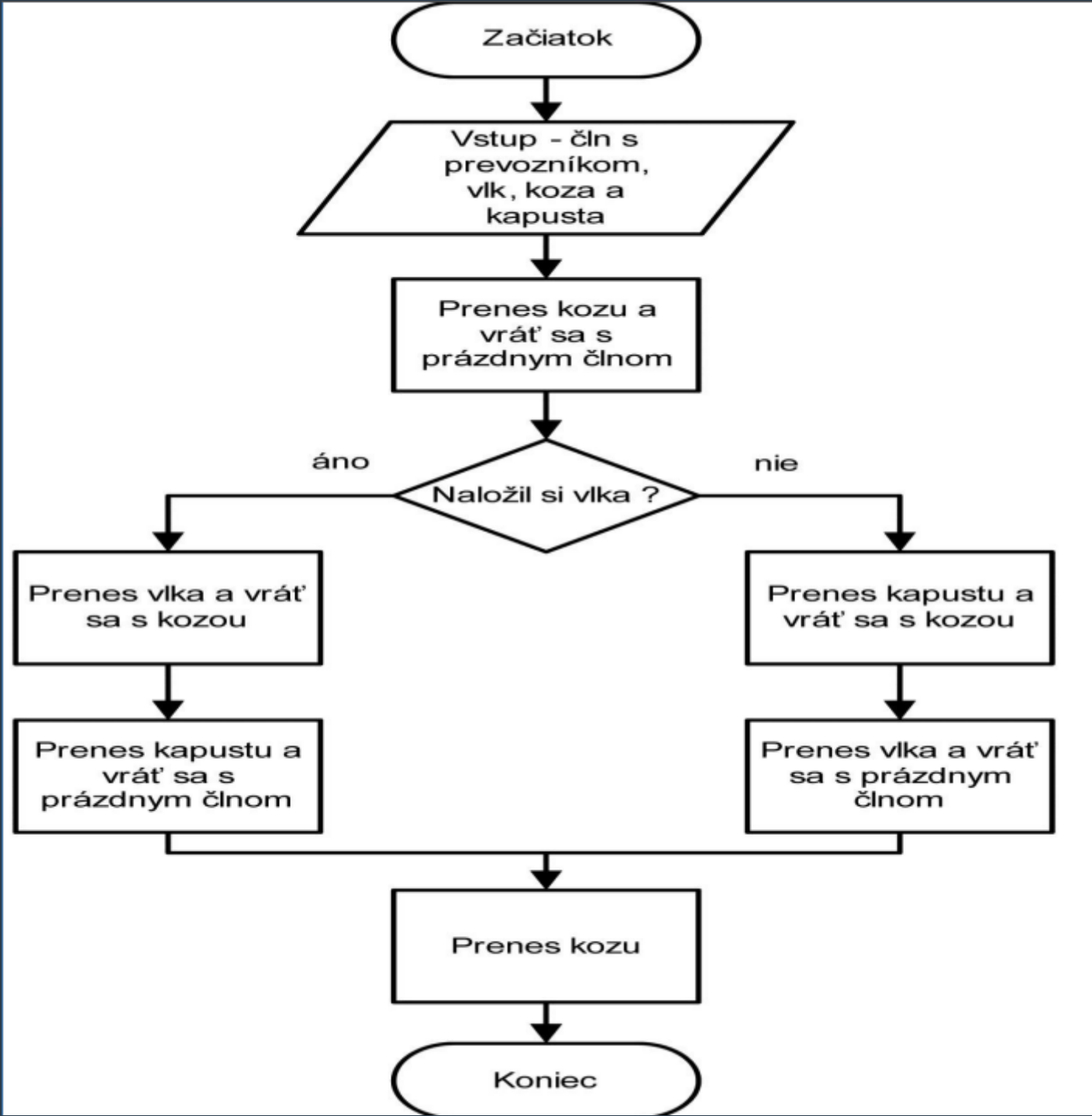
- Prievozník má cez rieku previezť kozu, vlka a kapustu.
- K dispozícii má jeden čln, do ktorého sa okrem neho zmestí ešte jeden „pasažier“.

Úloha: Akým spôsobom splní úlohu, ak nesmie nechať na jednom brehu kozu s kapustou a vlka s kozou?

Pri riešení úlohy si treba uvedomiť, že prievozník môže prevážať „pasažiera“ nielen tam, ale aj naspäť a že vlk nie je vegetarián, preto môže zostať na brehu s kapustou.

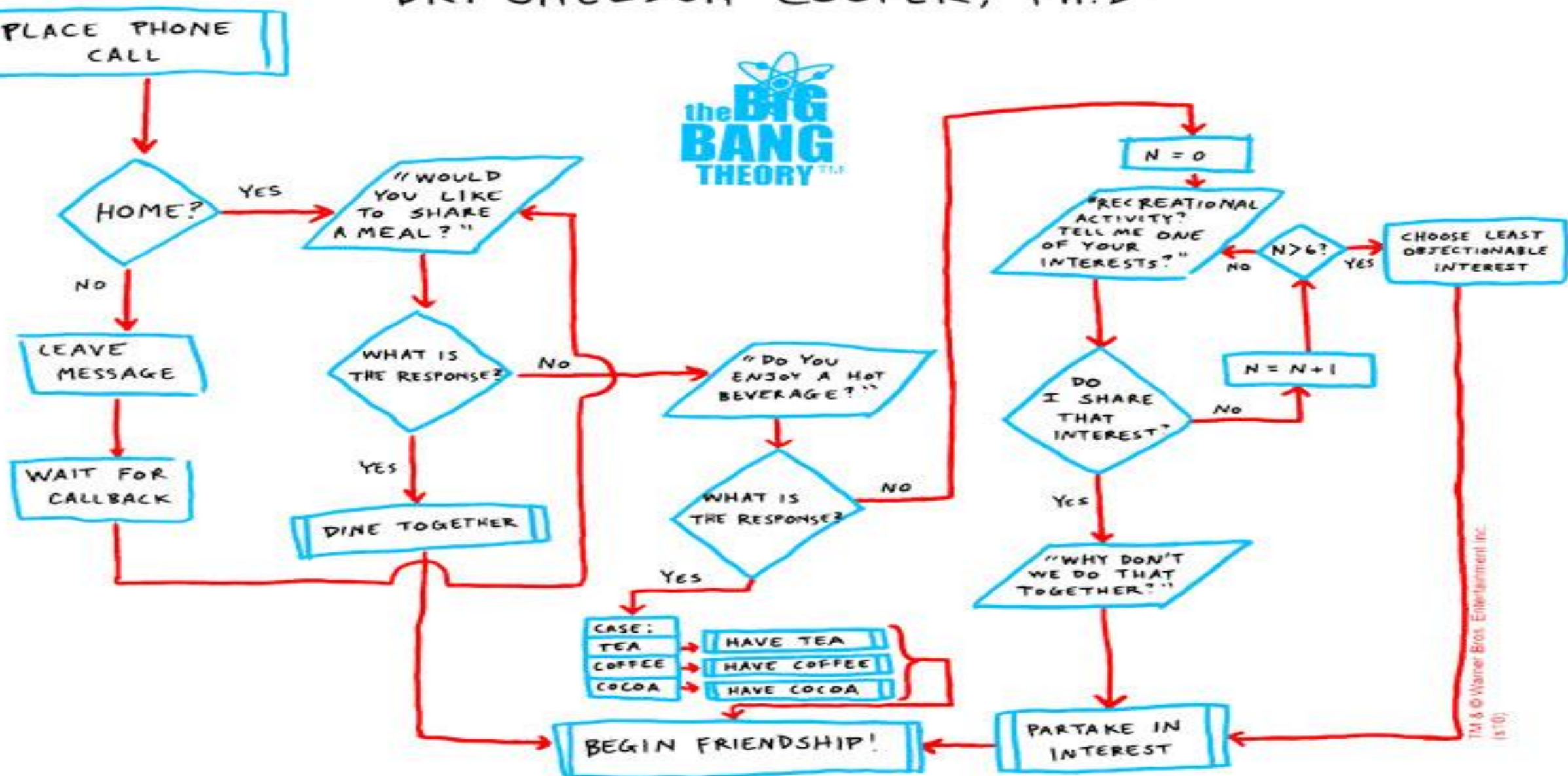
Úlohu môžeme zapísať v nasledujúcich krokoch:

- 1. krok: Prievozník prevezie na druhý breh kozu.
- 2. krok: Vráti sa s prázdny člnom.
- 3. krok: Na druhú stranu prevezie kapustu alebo vlka (má možnosť voľby).
- 4. krok: Vráti sa s kozou, ktorú vyloží.
- 5. krok: Prevezie vlka (ak v predchádzajúcom kroku previezol kapustu) alebo kapustu.
- 6. krok: Vráti sa s prázdny člnom.
- 7. krok: Prevezie kozu.



THE FRIENDSHIP ALGORITHM

DR. SHELDON COOPER, Ph.D



Príklad na zamyslenie

Kanibali a misionári

- Traja misionári a traja kanibali sa stretli na jednom brehu rieky. Na brehu bola malá loďka, na ktorú sa zmestia maximálne dvaja.
- Všetci by sa chceli prepraviť na druhý breh, ale na žiadnom brehu nesmie nikdy zostať prevaha kanibalov nad misionármi, inak by mohlo dôjsť k tragédií.

**DAKUJEM ZA
POZORNOST!**

