Analisi e Progetto di Algoritmi

Esercitazioni- Cozzi

1. HCS
2. LCS3
3. LICS
4. LCS R simboli rossi
5. Distanza di Edit
6. LCS < L
7. KS rossi
8. LCS El. Consecutivi
9. Problema di decisione Subset Sum
10. FW es vertici consecutivi rossi
11. FW min vertici consecutivi rossi
12. FW es archi consecutivi rossi
13. FW min archi consecutivi rossi
14. FW <= L
15. FW min archi rossi e = L
16. FW min pari archi rossi
17. FW min archi tutti colori

Definizione: Abbiamo due sequenze X, Y e una funzione $W : \Sigma \rightarrow \mathbb{N}$ funzione che associa ad ogni elemento di X e Y un peso. Vogliamo massimizzare il peso.

Sottoproblema (i,j): contiene il peso massimo della sottosequenza comune Z considerando i primi X_i e Y_j elementi

Equazioni di ricorrenza

- caso base: $M[i,j] = 0$ se $i = 0 \vee j = 0$
- passo ricorsivo: $M[i,j] = \begin{cases} M[i-1, j-1] + w[x_i] & \text{se } x[i] = y[j] \\ \max\{M[i-1, j], M[i, j-1]\} & \text{se } x[i] \neq y[j] \end{cases}$

Osservazione: Il problema è identico alla LCS, ma al posto di massimizzare la lunghezza, voglio massimizzare il peso.

Osservazione: Tutti i problemi di PD su sequenze che incontreremo possono essere formulati in termini:

- LCS: mi basta salvarmi il valore migliore nell'ultima posizione
- LIS: devo guardare sempre tutte le possibilità. Devo introdurre un problema ausiliario in cui salvarmi la sequenza che termina con X_m.

Definizione problema: date 3 sequenze X, Y, W di n, m e l numeri interi, si determini una tra le più lunghe sottosequenze comuni a X, Y e W
- sol. ottimale: S
- valore ottimo: $|S|$

Sottoproblema: Il sottoproblema di dimensione (i, j, k) è definito come segue: date 3 sequenze X, Y, W di n, m e l numeri interi, si determini una tra le più lunghe sottosequenze comuni ai prefissi X_i, Y_j, W_k

Numero sottoproblemi: Dato che $0 \le i \le n$, $0 \le j \le m$, $0 \le k \le l$ abbiamo $(n+1)(m+1)(l+1)$ sottoproblemi

Variabile sottoproblema: $c_{i,j,k}$ è la lunghezza di una tra le più lunghe sottosequenze comuni a X_i, Y_j, W_k

Equazioni di ricorrenza: si scrivono senza sapere il valore delle soluzioni dei sottoproblemi ma sapendo solamente che tali soluzioni esistono e si possono utilizzare. Composta da:
- caso base: (i, j, k) con $i = 0 \vee j = 0 \vee k = 0$
- passo ricorsivo: (i, j, k) con $i > 0 \wedge j > 0 \wedge k > 0$. Distinguiamo i casi:

$$c_{i,j,k} = \begin{cases} 1 + c_{i-1,j-1,k-1} & \text{se } x_i = y_j = w_k \\ \max\{c_{i-1,j,k}, c_{i,j-1,k}, c_{i,j,k-1}\} & \text{altrimenti} \end{cases}$$

Soluzione: Otteniamo il valore ottimo in posizione $c_{n,m,l}$, per ricostruire la soluzione ottimale salvando gli elementi e stampandoli tramite procedura ricorsiva.

Implementazione bottom-up: valore ottimo

```
* inizializzazione a 0 ∀ i,j,k tc i∨j∨k=0 *
for i=1 to n
    for j=1 to m
        for k=1 to l
            if xi=yj=wk
                c[i,j,k] = 1 + c[i-1,j-1,k-1]
            else
                c[i,j,k] = max{c[i-1,j,k], c[i,j-1,k], c[i,j,k-1]}
return c[n,m,l]
```

LICS - Valore ottimo

Definizione: Date due sequenze $X = \langle x_1, \dots, x_m \rangle$ e $Y = \langle y_1, \dots, y_n \rangle$ trovare la sottosequenza comune crescente più lunga.

- Valore ottimo: $|W|$
- Soluzione ottimale:

Definizione sottoproblema: trovare la lunghezza di una delle LICS di X_i e Y_j che terminano con x_i e y_j. N° sottoproblemi: $(m+1) \cdot (n+1)$ ↓ ottimizzabile in $m \cdot n$, avremo $c_{i,j}$ = lunghezza LICS x_i e y_j.

Equazioni di ricorrenza:

- caso base: $i = 0 \wedge j = 0$ tc $x_i \neq y_j \longrightarrow c_{i,j} = 0$
- passo ricorsivo: $i > 0 \wedge j > 0$ tc $x_i = y_j$ avremo

$$c_{i,j} = 1 + \max\{c_{h,k} \mid 1 \leq h < i,\ 1 \leq k < j,\ x_h < x_i\}$$ se $c_{h,k} = \emptyset$ allora $\max \emptyset = 0$

Soluzione del problema: ripercorro la matrice e salvo il $\max\{c_{i,j} \mid 1 \leq i \leq m,\ 1 \leq j \leq n\}$

	2	7	4	23	21	14	1	8
2	1	0	0	0	0	0	0	0
4	0	0	2	0	0	0	0	0
7	0	2	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
21	0	0	0	0	3	0	0	0
14	0	0	0	0	0	3	0	0
1	0	0	0	0	0	1	0	0

Algoritmo DP (bottom-up) valore ottimo:

```
max = 0
for i = 1 to m
    for j = 1 to n
        if xi ≠ yj
            c[i,j] = 0
        else
            temp = 0
            for h = 1 to i-1
                for k = 1 to j-1
                    if xh < xi ∧ c[h,k] > temp
                        temp = c[h,k]
            c[i,j] = 1 + temp
        if c[i,j] > max
            max = c[i,j]
return max
```

tempo: $\Theta(n^2 \cdot m^2)$

spazio: $\Theta(n \cdot m)$

Premessa: Sia C insieme di colori tc la funzione di colorazione è definita nel seguente modo

$$\phi(x): \begin{cases} \text{rosso} & x < 5 \\ \text{blu} & 5 \leq x \leq 10 \\ \text{verde} & x > 10 \end{cases}$$

Definizione: Date due sequenze X e Y di m e n numeri interi, e un numero naturale R stabilire se tutte le LCS di X e Y contengono al più R elementi rossi. (LCS < R)

Sottoproblema: Il sottoproblema di dimensione (i,j,r) è definito come segue:
Date due sequenze X_i e Y_j di m e n numeri interi, e un numero naturale R stabilire se tutte le LCS di X e Y contengono al più r elementi rossi.
N° sottoproblemi: $(m+1)(n+1)(R+1)$. Ad ogni sottoproblema è associata una variabile

Variabile: $c_{i,j,r}$ = true sse tutte le LCS a X_i e Y_j contengono al più r elementi rossi, false altrimenti

Equazioni di ricorrenza:

caso base: $c_{i,j,r} = \begin{cases} \text{true} & \text{se } i=0 \lor j=0 \\ \text{false} & \text{se } i>0 \land j>0 \land r=0 \land x_i = y_j \land \phi(x_i) = \text{rosso} \end{cases}$

passo ricorsivo: $c_{i,j,r} = \begin{cases} c_{i-1,j-1,r-1} & \text{se } i>0 \land j>0 \land r=0 \land x_i = y_j \land \phi(x_i) = \text{rosso} \\ c_{i-1,j-1,r} & \text{se } i>0 \land j>0 \land r\geq 0 \land x_i = y_j \land \phi(x_i) \neq \text{rosso} \\ c_{i-1,j,r} \land c_{i,j-1,r} & \text{se } i>0 \land j>0 \land r\geq 0 \land x_i = y_j \land \phi(x_i) = \text{rosso} \end{cases}$

Soluzione del problema: $c_{m,n,R}$

Algoritmo DP:

```
procedure LCSR(X,Y,R)
    for i=0 to m
        for r=0 to R
            c[i,0,r] = true
    for j=0 to m
        for r=0 to R
            c[0,j,r] = true
    for i=1 to m
        for j=1 to n
            if xi=yj and ϕ(xi) = rosso
                c[i,j,0] = false
    for i=1 to m
        for j=1 to n
            for r=0 to R
                if xi=yj
                    if ϕ(xi) = rosso
                        if r>0
                            c[i,j,r] = c[i-1,j-1,r-1]
                    else
                        C[i,j,r] = C[i-1,j-1,r]
                else
                    c[i,j,r] = c[i-1,j,r] ∧ c[i,j-1,r]
    return c[m,n,R]
```

Definizione: La distanza di Edit di due sequenze X e Y è il minimo numero di operazioni di cancellazione, inserimento e sostituzione di un singolo elemento che trasformano X in Y

Sottoproblema (i,j): trovare la distanza di edit dei prefissi X_i e Y_j, $i \in \{0,1,\dots,m\}$, $j \in \{0,1,\dots,n\}$
N° sottoproblemi: $(m+1)(n+1)$
Coefficiente $d_{i,j}$: distanza di edit dei prefissi X_i e Y_j
Valore ottimo: $d_{m,n}$

Casi base: sottoproblemi più semplici risolvibili
- $i=0 \wedge j=0$ allora sono uguali, $d_{0,0}=0$
- $i>0 \wedge j=0$ allora elimino gli i, $d_{i,0}=i$
- $i=0 \wedge j=0$ allora elimino gli j, $d_{0,j}=j$

Passo ricorsivo:
$$d_{i,j} = \begin{cases} d_{i-1,j-1} & \text{se } x_i = y_j \\ \min(\underbrace{d_{i-1,j-1}+1}_{\text{sostituzione}}, \underbrace{d_{i-1,j}+1}_{\text{cancellazione}}, \underbrace{d_{i,j-1}+1}_{\text{inserimento}}) & \text{se } x_i \neq y_j \end{cases}$$

Algoritmo DP:
```
int calcola_ED(X,Y)
    for i=0 to m
        D[i,0]=i
    for j=0 to n
        D[0,j]=j
    for i=1 to m
        for j=1 to n
            if xi=yj
                D[i,j]=D[i-1,j-1]
            else
                D[i,j]=min(D[i-1,j-1]+1, D[i-1,j]+1, D[i,j-1]+1)
    return D[m,n]
```

Ricostruzione operazioni:

- Inizializzo lista minOP vuota
- Si parte dalla cella D[m,n] e seguo percorso di calcolo e termino in D[0,0]
- Per ogni D[i,j] visitata si aggiunge operazione a minOP (se $x_i \neq y_j$)
- Da D[i,j] si passa alla cella che ne ha determinato il calcolo

Ricostruzione sol. ottimale:

```
void    ric_minop_ricorsiva(D, i, j)
    if i>0 or j>0
        if from diagonal
            if xi ≠ yj
                append substitution(xi → yj) to minOP
            ric_minop_ricorsiva(D, i-1, j-1)
        else if from left
            append insertion(yj) to minOP
            ric_minop_ricorsiva(D, i, j-1)
        else
            append deletion(yj) to minOP
            ric_minop_ricorsiva(D, i-1, j)
```

Esempio: $X = \langle 2, 10, 5, 3, 1, 12 \rangle$, $Y = \langle 2, 5, 12, 2, 3, 12, 1, 30 \rangle$

		2	5	12	2	3	12	1	30
	0	1	2	3	4	5	6	7	8
2	1	0	1	2	3	4	5	6	7
10	2	1	1	2	3	4	5	6	7
5	3	2	1	2	3	4	5	6	7
3	4	3	2	2	3	3	4	5	6
1	5	4	3	3	3	4	4	4	5
12	6	5	4	3	4	4	4	5	5

		2	5	12	2	3	12	1	30
	0	1	2	3	4	5	6	7	8
2	1	0	1	2	3	4	5	6	7
~~10~~	2	1	1	2	3	4	5	6	7
5 ↪ 2 ↪ 12	3	2	1	2	3	4	5	6	7
3 ↪ 12	4	3	2	2	3	3	4	5	6
1	5	4	3	3	3	4	4	4	5
12 → 30	6	5	4	3	4	4	4	5	5

Definizione: Date due sequenze X e Y rispettivamente di m e n interi e un numero naturale L, stabilire se la lunghezza di una qualunque LCS tra X e Y sia $\leq$ a L.

Sottoproblema (i,j,ℓ): Date due sequenze X e Y rispettivamente di m e n interi e un numero naturale L, stabilire se la lunghezza di una qualunque LCS tra X_i e Y_j sia $\leq$ a ℓ.

N° sottoproblemi: Dato che $0 \leq i \leq m$, $0 \leq j \leq n$, $0 \leq \ell \leq L$ avremo $(m+1)(n+1)(L+1)$ sottoproblemi

Definizione coefficiente: $c_{i,j,\ell}$ é true se e solo se la LCS X_i e Y_j é $\leq \ell$, false altrimenti.

Equazioni di ricorrenza:

caso base
$$\begin{cases} \text{true} & \text{se } i=0 \lor j=0 \\ \text{false} & \text{se } i>0 \land j>0 \land j=0 \land x_i=y_j \end{cases}$$

passo ricorsivo
$$\begin{cases} c_{i-1,j-1,\ell-1} & \text{se } i>0 \land j>0 \land \ell>0 \land x_i=y_j \\ c_{i-1,j,\ell} \land c_{i,j-1,\ell} & \text{se } i>0 \land j>0 \land \ell\geq 0 \land x_i \neq y_j \end{cases}$$

Algoritmo DP: procedure LCS≤L(X, Y, L)

```
for i=0 to m
    for ℓ=0 to L
        c[i,0,ℓ]=true

for j=0 to m
    for ℓ=0 to L
        c[0,j,ℓ]=true

for i=1 to m
    for j=1 to n
        if xi=yj
            c[i,j,0]=false

for i=1 to m
    for j=1 to n
        for ℓ=0 to L
            if xi=yj
                if ℓ>0
                    c[i,j,ℓ]=c[i-1,j-1,ℓ-1]
            else
                c[i,j,ℓ]=c[i,j-1,ℓ] ∧ c[i-1,j,ℓ]
return c[m,n,L]
```


Input:

- $X = \{1, 2, \dots, n\}$ insieme di n oggetti dove ogni oggetto i ha associato:
  - v_i, valore
  - w_i, peso
  - $col(i)$, colore nell'insieme {rosso, blu}
- C, capacità massima dello zaino
- R, intero positivo

Output:

- Problema: subset S di X che ha il massimo valore che contenga al più R oggetti rossi.
- Problema ridotto: valore del subset S di X che ha il massimo valore che contenga al più R oggetti rossi.

Sottoproblema (i, c, r): trovare il valore del subset $S_{i,c,r}$ dell'insiemi $X_i = \{1, 2, \dots, i\}$ che possa essere contenuto in uno zaino di capacità c, che ha il massimo valore e che contenga al più r oggetti rossi.

Numero sottoproblemi: $0 \leq i \leq n$, $0 \leq c \leq C$, $0 \leq r \leq R$ quindi $(m+1)(C+1)(R+1)$ sottoproblemi

Definizione coefficiente: $d_{i,c,r}$ contiene il valore di $S_{i,c,r}$

Equazioni di ricorrenza:

- caso base: $d_{i,c,r} = 0$, $S_{i,c,r} = \emptyset$ se $(i = 0 \lor c = 0) \land r \geq 0$
- passo ricorsivo:

$$
\begin{cases}
d_{i,c,r} = d_{i-1,c,r} & \text{se } i>0 \land c>0 \land r \geq 0 \land w_i > c \\
d_{i,c,r} = d_{i-1,c,r} & \text{se } i>0 \land c>0 \land r = 0 \land w_i \leq c \land col(i) = R \\
d_{i,c,r} = \max(d_{i-1,c,r} \land d_{i-1,c-w_i,r-1} + v_i) & \text{se } i>0 \land c>0, r>0, col(i) = R \\
d_{i,c,r} = \max(d_{i-1,c,r} \land d_{i-1,c-w_i,r} + v_i) & \text{se } i>0 \land c>0, r>0, col(i) \neq R
\end{cases}
$$

Definizione: Date due sequenze X e Y di lunghezza m e n trovare la LCS che non ha 2 elementi consecutivi uguali.

Problema ausiliario: Date due sequenze X e Y di lunghezza m e n trovare la LCS che non ha 2 elementi consecutivi uguali e finisce in x_m e y_n.

Sottoproblema (i,j): Date due sequenze X_i e Y_j di lunghezza m e n trovare la LCS che non ha 2 elementi consecutivi uguali e finisce in x_i e y_j.

Numero sottoproblemi: Dato che $0 \le i \le m$ e $0 \le i \le n$ avremo $(n+1)(m+1)$ sottoproblemi.

Definizione coefficiente: $c_{i,j}$ contiene la lunghezza della LCS di X_i e Y_j che non ha 2 elementi consecutivi e finisce in x_i e y_j.

Equazioni di ricorrenza:

- caso base: $c_{i,j} = 0$ se $(i=0 \lor j=0) \lor (i>0 \land j>0, x_i \neq y_j)$
- passo ricorsivo: $c_{i,j} = 1 + \max\{c_{h,k} \neq 0 \mid tc\ 1 \le h < i \land 1 \le k \le j \land x_h \neq x_i\}$

Soluzione problema ausiliario: $c_{m,n}$

Soluzione problema ridotto: $\max\{c_{i,j} \mid 0 \le i \le m, 0 \le j \le n\}$

Subset sum - Problema di Decisione

Introduzione: Prima di introdurre il problema Subset Sum, introduciamo il Partition, caso specifico del SS.

Definizione: Abbiamo $O = (v_1, v_2, \dots, v_n)$ di n oggetti di valore v_i. Vogliamo sapere se é possibile o meno dividere equamente i valori.

Def. formale: $O = \{o_1, \dots, o_n\}$ oggetti, $V = \{v_1, \dots, v_n\}$ valori, $? \exists S \subseteq O$ tc $\sum_{i \in S} v_i = \sum_{i \in O \cdot S} v_i$

Idea: Faccio una somma dei valori $T = \sum_{i \in O} v_i$ e la divido per 2: $T/2$. Se T é dispari allora la risposta é no, se T é pari devo scegliere un sottoinsieme di oggetti che arrivano a $T/2$.

Problema: Come faccio a sapere se, preso un numero, devo prenderne anche un altro? Dovrei provare tutte le combinazioni: tempo esponenziale (2^n). Allora cerco algoritmo DP.

Idea: Creo una matrice di n righe e $T/2$ colonne, dove in posizione $M[i][j] = T/F$, ed é T sse posso ottenere il valore j usando i primi i oggetti.

Sottostruttura ottima: $m_{i,j} = \begin{cases} T & \text{se } j \text{ é ottenibile dai primi } i \\ F & \text{se } j \text{ non é ottenibile dai primi } i \end{cases}$

Equazioni di ricorrenza:

- caso base: $M[0,j] = F \quad \forall 1 \leq j \leq T/2$
- passo ricorsivo: $M[i,j] = \begin{cases} T & \text{se } j = v_i \\ M[i-1,j] & \text{se } j < v_i \\ M[i-1,j] \vee M[i-1,j-v_i] & \text{se } j > v_i \end{cases}$
- valore ottimo: $M[n][T/2]$

Algoritmo bottom-up:

```
bool partition_DP(n, V)
    T = 0
    for i = 1 to n
        T = T + V[i]
    if T mod 2 ≠ 0
        return false
    else
        for j = 1 to T/2
            M[0, j] = F
        for i = 1 to T/2
            for j = 1 to n
                if V[i] == j
                    M[i, j] = T
                else if V[i] > j
                    M[i, j] = M[i-1, j]
                else
                    M[i, j] = M[i-1, j] or M[i-1, j-V[i]]
    return M[n, T/2]
```

Tempo: $O(n \cdot T/2)$ pseudopolinomiale. Possiamo ottimizzare il caso medio con un controllo: appena vedo T in colonna T/2 restituisco e termino.

Spazio: $O(n \cdot T/2)$ ottimizzabile in $O(T)$ dato che bastano due righe.

Subset sum - Problema di Decisione

Definizione: $O = \{1, \dots, n\}$ oggetti, $V = \{v_1, \dots, v_n\}$, L limite, $? \exists S \subseteq O$ tc $\sum_{i \in S} v_i = L$

Idea: Posso sfruttare partition usando al posto di $T/2 \rightarrow L$.

Algoritmo:

```
Partition(O,V):
    T=0
    for i=1 to n
        T=T+V[i]
    if T mod 2 ≠ 0
        return false
    else
        R= subsetsum(O,V,T/2)
    return R
```

Terminologia: Abbiamo ridotto partition a subsetsum. Riduzione di problemi.

Definizione: Dato un grafo $G=(V,E,col)$ dove V sono i vertici, E sono gli archi, $col: V \rightarrow \{R,N\}$ si vuole determinare $\forall (i,j) \in V \times V$ l'esistenza di un cammino p da i in j in cui non ci sono due vertici consecutivi rossi.

Sottoproblema K-esimo: Dato un grafo $G=(V,E,col)$ e $K \in \{0,1,\dots,n\}$ si vuole determinare $\forall (i,j) \in V \times V$ l'esistenza di un cammino p da i a j senza vertici consecutivi rossi.

Variabile associata: $d^{(K)}_{(i,j)}$: True se esiste cammino dal vertice i al vertice j i cui vertici intermedi appartengono all'insieme $\{1,2,\dots,K\}$ e che non contiene due vertici consecutivi rossi.

Equazioni di ricorrenza:

caso base: $\forall (i,j) \in V \times V$, $d^{(0)}_{(i,j)} = \begin{cases} \text{True} & \text{se } i=j \lor (i,j) \in E \land [col[i] \neq R \lor col[j] \neq R] \\ \text{False} & \text{altrimenti} \end{cases}$

passo ricorsivo: $d^{(K)}_{(i,j)} = d^{(K-1)}_{(i,j)} \lor d^{(K-1)}_{(i,K)} \land d^{(K-1)}_{(K,j)}$

Algoritmo DP:

```
procedure FW-no-red-pair-ex(V, E, col())
    for i=1 to n
        for j=1 to n
            if i=j ∨ (i,j)∈E ∧ [col[i]≠R ∨ col[j]≠R]
                d^(0)(i,j) = True
            else
                d^(0)(i,j) = False
    for K=1 to n
        for i=1 to n
            for j=1 to n
                d^(K)(i,j) = d^(K-1)(i,j) ∨ d^(K-1)(i,K-1) + d^(K-1)(K,j)
    return d^(n)
```

Tempo: (n^3)

Definizione: Dato un grafo $G=(V,E,W,col)$ dove V sono i vertici, E sono gli archi, W è la matrice dei pesi, $col: V \rightarrow \{R,N\}$ si vuole calcolare $\forall (i,j) \in V \times V$ il peso di un cammino minimo p da i a j in cui non ci siano due vertici consecutivi rossi.

Sottoproblema K-esimo: Dato un grafo $G=(V,E,W,col)$ con $K \in \{0,1,\dots,n\}$ si vuole calcolare $\forall (i,j) \in V \times V$ il peso di un cammino minimo p da i a j con vertici intermedi $\{1,2,\dots,K\}$ in cui non ci siano due vertici consecutivi di colore rosso.

Definizione variabile: $d^{(K)}(i,j)$ che è il peso di un cammino minimo da i a j i cui vertici intermedi appartengono all'insieme $\{1,2,\dots,K\}$ e che non contiene due vertici consecutivi di colore rosso.

Equazioni di ricorrenza:

caso base: $\forall (i,j) \in V \times V$, $d_{i,j}^{(0)} = \begin{cases} 0 & \text{se } i=j \\ w_{i,j} & \text{se } i \neq j \wedge (i,j) \in E \wedge [col(i) \neq R \vee col(j) \neq R] \\ \infty & \text{altrimenti} \end{cases}$

passo ricorsivo: $\forall (i,j) \in V \times V$, $d_{i,j}^{(K)} = \min\{ \underset{K \notin p}{d^{(K-1)}(i,j)}, \underset{K \in p}{d^{(K-1)}(i,K) + d^{(K-1)}(K,j)} \}$ per proprietà sottostruttura ottima

Algoritmo DP:

```
procedure FW-no-red-pair(V, E, col())
    for i=1 to n
        for j=1 to n
            if i=j  d(0)(i,j)=0
            else if (i,j)∈E ∧ [col(i)≠R ∨ col(j)≠R}  d(0)(i,j)=w_i,j
            else d(0)(i,j)=∞
    for K=1 to n
        for i=1 to n
            for j=1 to n
                d(K)(i,j)=min{d(K-1)(i,j), d(K-1)(i,K-1)+d(K-1)(K,j)}
    return d(n)
```

Tempo: (n^3)

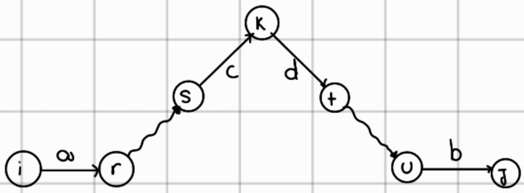
Definizione: Dato un grafo $G=(V,E,col)$ dove V sono i vertici, E sono gli archi, $col: E \rightarrow \{R,N\}$ si vuole determinare $\forall (i,j) \in V \times V$ l'esistenza di un cammino p da i in j in cui non ci sono due archi consecutivi rossi.

Sottoproblema K-esimo: Dato un grafo $G=(V,E,col)$ e $K \in \{0,1,\dots,n\}$ si vuole determinare $\forall (i,j) \in V \times V$ l'esistenza di un cammino p da i a j senza archi consecutivi rossi, con vertici intermedi $\in \{1,\dots,K\}$

Variabile associata: $d^{(K)}(i,j)$: True se esiste cammino dal vertice i al vertice j i cui vertici intermedi appartengono all'insieme $\{1,2,\dots,K\}$ e che non contiene due archi consecutivi rossi.

Introduzione p.ausiliario: Necessario introdurlo in quanto l'ultimo arco di p_1 e il primo arco di p_2 non vengono controllati.

P ausiliario p': Dato un grafo $G=(V,E,col)$ dove V sono i vertici, E sono gli archi, $col: E \rightarrow \{R,N\}$ si vuole determinare $\forall (i,j) \in V \times V$ e $\forall (a,b) \in \{R,N\} \times \{R,N\}$ l'esistenza di un cammino p da i in j dove il primo arco ha colore a e l'ultimo ha colore b



Sottoproblema K-esimo P: Dato un grafo $G=(V,E,col)$ e $K \in \{0,1,\dots,n\}$ si vuole determinare $\forall (i,j) \in V \times V$ e $\forall (a,b) \in \{R,N\} \times \{R,N\}$ l'esistenza di un cammino p da i a j senza archi consecutivi rossi.

Variabile associata: $d^{(K)}(i,j,a,b)$ vale True se esiste un cammino minimo dal vertice i al vertice j con vertici intermedi appartenenti all'insieme $\{1,2,\dots,K\}$ senza due archi consecutivi rossi, con il primo arco di colore a e l'ultimo arco di colore b. False altrimenti

Dimensione: $D^{(K)}(i,j,a,b) = n \cdot n \cdot 2 \cdot 2 = 4n^2$

Equazioni di ricorrenza:

- caso base: $K=0$ allora $d^{(0)}(i,j,a,b) = \begin{cases} \text{True} & \text{se } i \neq j \wedge (i,j) \in E \wedge a=b=col(i,j) \\ \text{False} & \text{altrimenti} \end{cases}$
- passo ricorsivo: $K>0$ allora $d^{(K)}(i,j,a,b) = e_1 \vee e_2$ dove:
  - se $K \notin p$ $e_1 = d^{(K-1)}(i,j,a,b)$
  - se $K \in p$ $e_2 = \bigvee_{(c,d) \in \{R,N\}^2 \mid c \neq R \vee d \neq R} (d^{(K-1)}(i,K,a,c) \wedge d^{(K-1)}(K,j,d,b))$

Soluzione problema PB: Chiamiamo $d_{PB}(i,j)$ il valore di verità del cammino $i \to j$.
Per ogni coppia (i,j) tc $i=j$ avremo $d_{PB}(i,j)=\text{True}$, mentre se $i \neq j$ avremo uno dei 4 casi:

- $v_1 = d^{(n)}(i,j,R,R)$
- $v_2 = d^{(n)}(i,j,R,N)$
- $v_3 = d^{(n)}(i,j,N,R)$
- $v_4 = d^{(n)}(i,j,N,N)$

Per ogni (i,j) il $d_{PB}(i,j)$ è dato dall'or logico tra v_1, v_2, v_3, v_4. La soluzione ha n^2 elementi.

Algoritmo DP:

```
procedure FW-no-cons-red-edges(V, E, col())
    for i = 1 to n                                         caso base problema ausiliario
        for j = 1 to n
            for a ∈ {R, B}
                for b ∈ {R, B}
                    if i ≠ j ∧ (i,j) ∈ E ∧ a = b = col(i,j)
                        d^(0)_{i,j,a,b} = True
                    else
                        d^0(i,j,a,b) = False
    for K = 1 to n                                         passo ricorsivo problema ausiliario
        for i = 1 to n
            for j = 1 to n
                for a ∈ {R, B}
                    for b ∈ {R, B}
                        e1 = d^(K-1)_{i,j,a,b}
                        e2 = VALORE(K, i, j, a, b)
                        d^(K)(i,j) = e1 ∨ e2
    for i = 1 to n                                         soluzione problema originale
        for j = 1 to n
            if i ≠ j
                u1 = d^(n)_{i,j,R,N}
                u2 = d^(n)_{i,j,N,N}
                u3 = d^(n)_{i,j,N,R}
                u4 = d^(n)_{i,j,R,R}
                dPB(i,j) = ∨{u1, u2, u3, u4}
            else
                dPB = True
    return dPB
```

Definizione: Dato un grafo $G=(V,E,W,col)$ dove V sono i vertici, E sono gli archi, W è la matrice dei pesi, $col: E \rightarrow \{R,N\}$ si vuole calcolare $\forall (i,j) \in V \times V$ il peso di un cammino minimo p da i a j in cui non ci siano due archi consecutivi rossi.

Sottoproblema K-esimo: Dato un grafo $G=(V,E,W,col)$ con $K \in \{0,1,\dots,n\}$ si vuole calcolare $\forall (i,j) \in V \times V$ il peso di un cammino minimo p da i a j con vertici intermedi $\{1,2,\dots,K\}$ in cui non ci siano due archi consecutivi di colore rosso.

Definizione variabile: $d^{(K)}(i,j)$ che è il peso di un cammino minimo da i a j i cui vertici intermedi appartengono all'insieme $\{1,2,\dots,K\}$ e che non contiene due archi consecutivi di colore rosso.

Introduzione p. ausiliario: Necessario introdurlo in quanto l'ultimo arco di p_1 e il primo arco di p_2 non vengono controllati.

P ausiliario p': Dato un grafo $G=(V,E,col)$ dove V sono i vertici, E sono gli archi, $col: E \rightarrow \{R,N\}$ si vuole calcolare $\forall (i,j) \in V \times V$ e $\forall (a,b) \in \{R,N\} \times \{R,N\}$ il peso minimo di un cammino p da i in j dove il primo arco ha colore a e l'ultimo ha colore b

Sottoproblema K-esimo P': Dato un grafo $G=(V,E,col)$ e $K \in \{0,1,\dots,n\}$ si vuole calcolare $\forall (i,j) \in V \times V$ e $\forall (a,b) \in \{R,N\} \times \{R,N\}$ il peso minimo di un cammino p da i a j senza archi consecutivi rossi.

Variabile associata: $d^{(K)}(i,j,a,b)$ è il peso di un cammino minimo dal vertice i al vertice j con vertici intermedi appartenenti all'insieme $\{1,2,\dots,K\}$ senza due archi consecutivi rossi, con il primo arco di colore a e l'ultimo arco di colore b, False altrimenti

Equazioni di ricorrenza:

- caso base: $K=0$ allora $d^{(0)}(i,j,a,b) = \begin{cases} \omega_{i,j} & \text{se } i \neq j \wedge (i,j) \in E \wedge a=b=col(i,j) \\ \infty & \text{altrimenti} \end{cases}$

- passo ricorsivo: $K>0$ allora $d^{(K)}(i,j,a,b) = e_1 \vee e_2$ dove:
  - se $K \notin p$ $e_1 = d^{(K-1)}(i,j,a,b)$
  - se $K \in p$ $e_2 = \min\limits_{(c,d) \in \{R,N\}^2 \mid c \neq R \vee d \neq R} (d^{(K-1)}(i,K,a,c) + d^{(K-1)}(K,j,d,b))\}$

Soluzione problema PB: Chiamiamo $d_{PB}(i,j)$ il peso del cammino $i \to j$.
Per ogni coppia (i,j) tc $i=j$ avremo $d_{PB}(i,j)=0$, mentre se $i \neq j$ avremo uno dei 4 casi:

- $v_1 = d^{(n)}(i,j,R,R)$
- $v_2 = d^{(n)}(i,j,R,N)$
- $v_3 = d^{(n)}(i,j,N,R)$
- $v_4 = d^{(n)}(i,j,N,N)$

Per ogni (i,j) il $d_{PB}(i,j)$ è dato dal minimo tra v_1, v_2, v_3, v_4. La soluzione ha n^2 elementi.

Algoritmo per minimo:

```
procedure minimo(K, i, j, a, b)
    m = ∞
    for c ∈ {R,B}
        for d ∈ {R,B}
            if c ≠ R v d ≠ R
                x = d^(K-1)(i,K,a,c) + d^(K-1)(K,j,d,b)
                if x < m
                    m = x
    return m
```

FW min archi rossi

Algoritmo DP:

```
procedure FW-no-cons-red-edges(V, E, col())
    for i = 1 to n                                   // caso base problema ausiliario
        for j = 1 to n
            for a ∈ {R, B}
                for b ∈ {R, B}
                    if i ≠ j ∧ (i,j) ∈ E ∧ a = b = col(i,j)
                        d^(0) i,j,a,b = w i,j
                    else
                        d^0(i,j,a,b) = ∞
    for K = 1 to n                                   // passo ricorsivo problema ausiliario
        for i = 1 to n
            for j = 1 to n
                for a ∈ {R, B}
                    for b ∈ {R, B}
                        e1 = d^(K-1) i,j,a,b
                        e2 = MINIMO(K, i, j, a, b)
                        d^(K)(i,j) = min{e1, e2}
    for i = 1 to n                                   // soluzione problema originale
        for j = 1 to n
            if i ≠ j
                U1 = d^(n) i,j,R,N
                U2 = d^(n) i,j,N,N
                U3 = d^(n) i,j,N,R
                U4 = d^(n) i,j,R,R
                dPB(i,j) = min{U1, U2, U3, U4}
            else
                dPB = 0
    return dPB
```

Definizione: Dato un grafo $G=(V,E,W)$ pesato, orientato e senza cappi e dato un intero $L>0$ calcolare $\forall (i,j) \in V \times V$ il peso di un cammino minimo da i a j di lunghezza $\leq L$.

Definizione sottoproblema: Dato un grafo $G=(V,E,W)$ e $k \in \{0,\dots,n\}$ e $\ell \in \{1,\dots,L\}$ $\forall (i,j)$ calcolare il peso del cammino minimo da i a j con vertici intermedi in $\{1,\dots,k\}$ e lunghezza $\leq L$.

Definizione variabili: $d_{i,j}^{(k,\ell)}$ è il peso del cammino minimo da i a j con vertici intermedi $\in \{1,\dots,k\}$ e lunghezza $\leq L$

Equazioni di ricorrenza:

caso base: $k=0$ $$d_{i,j}^{(0,\ell)} = \begin{cases} 0 & \text{se } i=j \\ w_{i,j} & \text{se } i \neq j \wedge (i,j) \in E \\ \infty & \text{altrimenti} \end{cases}$$

passo ricorsivo: $k>0$ $$d_{i,j}^{(k,\ell)} = \begin{cases} d_{i,j}^{(k-1,\ell)} & \text{se } k \notin p \\ \min\{ d_{i,k}^{(k-1,\ell_1)} + d_{k,p}^{(k-1,\ell_2)} \} \ \ell_1+\ell_2 \leq \ell & \text{se } k \in p \wedge \ell > 1 \\ \infty & \text{se } k \in p \wedge \ell = 1 \end{cases}$$

FW min archi rossi + $\ell = L$

Definizione: Dato un grafo $G=(V,E,W,col)$ dove V sono i vertici, E sono gli archi, W è la matrice dei pesi, $col: W \rightarrow \{R,N\}$ si vuole calcolare $\forall (i,j) \in V \times V$ il ~~peso di un cammino~~ minimo p da i a j in cui non ci siano due vertici consecutivi rossi e lunghezza $= L$

Sottoproblema K-esimo: Dato un grafo $G=(V,E,W,col)$ con $K \in \{0,1,\dots,n\}$ e $\ell \in \{1,\dots,L\}$ $\forall (i,j) \in V \times V$ il peso di un cammino minimo p da i a j con vertici intermedi $\{1,2,\dots,K\}$ in cui non ci siano due vertici consecutivi di colore rosso e lunghezza $= \ell$

Definizione variabile: $d^{(K,\ell)}(i,j)$ che è il peso di un cammino minimo da i a j i cui vertici intermedi appartengono all'insieme $\{1,2,\dots,K\}$ e che non contiene due vertici consecutivi di colore rosso e la lunghezza $\ell = L$

Equazioni di ricorrenza:

caso base $K=0$ avremo
$$d^{(0,\ell)}_{i,j} = \begin{cases} 0 & \text{se } i=j \\ w_{i,j} & \text{se } i \neq j \wedge (i,j) \in E \wedge [col(i) \neq R \vee col(j) \neq R] \wedge \ell = 1 \\ \infty & \text{altrimenti} \end{cases}$$

passo ricorsivo: $K>0$ avremo $d^{(K,\ell)}_{i,j} = \min\{e_1 \vee e_2\}$ tc

$$e_1 = d^{(K-1,\ell)}_{i,j} \qquad \text{se } K \notin p$$
$$e_2 = \min\{d^{K-1,\ell_1}_{i,K} + d^{K-1,\ell_2}_{K,j} \mid \ell_1 + \ell_2 = L \wedge \ell_1, \ell_2 \in \{1,\dots,\ell-1\}\} \qquad \text{se } K \in p$$

con $\min \emptyset = \infty$

Valore ottimo: $d^{(n,L)}(i,j)$

FW pari archi rossi

Istanza: (V, E, W, col) $col: E \to \{R, N, B, \dots\}$ finito

Soluzione: peso di un cammino minimo da i a j nel quale é pari il numero di archi rossi.

Sottoprob: (K, p) $K \in \{0, \dots, n\}$ $p \in \{0, 1\}$ $\forall (i,j) \in V^2$

Variabile: $d_{i,j}^{(K,p)}$: peso di un cammino minimo da i a j nel quale è pari/dispari il n° degli archi rossi a seconda che rispettivamente $p=0$ o $p=1$ e con vertici intermedi $\in \{1, \dots, K\}$

Caso base: $K=0$ $p \in \{0, 1\}$ $\forall (i,j) \in V^2$

$$d_{i,j}^{(0,0)} = \begin{cases} \emptyset & \text{se } i=j \\ w_{ij} & \text{se } i \neq j \wedge (i,j) \in E \wedge col(i,j) \neq R \\ \infty & \text{altrimenti} \end{cases} \qquad d_{i,j}^{(0,1)} = \begin{cases} \infty & \text{se } i=j \\ w_{ij} & \text{se } i \neq j \wedge (i,j) \in E \wedge col(i,j) = R \\ \infty & \text{altrimenti} \end{cases}$$

Passo ricorsivo: (K, p) $K>0$, $p \in \{0, 1\}$

se $K \notin$ cammino $d_{i,j}^{(K,p)} = d_{i,j}^{(K-1,p)} = e_1$

se $K \notin$ cammino

$$d_{i,j}^{(K,0)} = \min\{d_{iK}^{(K-1,0)} + d_{Kj}^{(K-1,0)},\ d_{iK}^{(K-1,1)} + d_{Kj}^{(K-1,1)}\} = e_2$$

$$d_{i,j}^{(K,1)} = \min\{d_{iK}^{(K-1,0)} + d_{Kj}^{(K-1,1)},\ d_{iK}^{(K-1,1)} + d_{Kj}^{(K-1,0)}\} = e_3$$

Equazioni di ricorrenza:

$$d_{ij}^{(K,0)} = \min\{e_1, e_2\}$$

$$d_{ij}^{(K,1)} = \min\{e_1, e_3\}$$

Soluzione: $\forall (i,j) \in V^2$ $d_{i,j}^{(n,0)}$

FW - archi - tutti - colori

Istanza: (V, E, W, col) $col: E \to \{R, B\}$

Soluzione: peso di un cammino minimo da i a j nel quale sono presenti entrambi i colori

Osservazione: rainbow-path è np-completo se $|C|>2$, tempo esponenziale.

Sottoprob: (K, p, q) $K \in \{0, \dots, n\}$, $p, q \in \{T, F\}$ $\forall (i,j) \in V^2$

Variabile: $d_{i,j}^{(K,p,q)}$ peso di un cammino minimo da i a j nel quale è presente un arco rosso sse $p=T$ e nel quale è presente un arco blu se $q=T$ e con vertici intermedi $\in \{1, \dots, K\}$

Caso base: (K, p, q) con $K=0 \wedge p \in \{T,F\}, q \in \{T,F\}$ $\forall (i,j) \in V^2$

$$d_{ij}^{(0,T,T)} = \infty \qquad d_{ij}^{(0,F,F)} = \begin{cases} 0 & \text{se } i=j \\ \infty & \text{altrimenti} \end{cases}$$

$$d_{ij}^{(0,T,F)} = \begin{cases} \infty & \text{se } i=j \\ w_{ij} & \text{se } i \neq j \wedge (i,j) \in E \wedge col(i,j)=B \\ \infty & \text{altrimenti} \end{cases} \qquad d_{ij}^{(0,F,T)} = \begin{cases} \infty & \text{se } i=j \\ w_{ij} & \text{se } i \neq j \wedge (i,j) \in E \wedge col(i,j)=R \\ \infty & \text{altrimenti} \end{cases}$$

Passo ricorsivo: (K, p, q) con $K>0$, $p, q \in \{T,F\}$ $\forall (i,j) \in V^2$

- $K \notin$ cammino $d_{i,j}^{(K,p,q)} = d_{i,j}^{(K-1,p,q)} = e_0$
- $K \in$ cammino abbiamo 4 casi:

$$d_{i,j}^{(K,T,T)} = \min_{r_1,r_2,b_1,b_2 \in \{T,F\}^4} \{ d_{iK}^{(K-1,r_1,b_1)} + d_{Kj}^{(K-1,r_2,b_2)} \} \text{ t.c. } (r_1=T \vee r_2=T) \wedge (b_1=T \vee b_2=T) = e_1$$

$$d_{i,j}^{(K,T,F)} = \min_{r_1,r_2,b_1,b_2 \in \{T,F\}^4} \{ d_{iK}^{(K-1,r_1,b_1)} + d_{Kj}^{(K-1,r_2,b_2)} \} \text{ t.c. } (r_1=T \vee r_2=T) \wedge (b_1=b_2=F) = e_2$$

$$d_{i,j}^{(K,F,T)} = \min_{r_1,r_2,b_1,b_2 \in \{T,F\}^4} \{ d_{iK}^{(K-1,r_1,b_1)} + d_{Kj}^{(K-1,r_2,b_2)} \} \text{ t.c. } (r_1=r_2=F) \wedge (b_1=T \vee b_2=T) = e_3$$

$$d_{i,j}^{(K,F,F)} = \min_{r_1,r_2,b_1,b_2 \in \{T,F\}^4} \{ d_{iK}^{(K-1,r_1,b_1)} + d_{Kj}^{(K-1,r_2,b_2)} \} \text{ t.c. } (r_1=r_2=F) \wedge (b_1=b_2=F) = e_4$$

Equazioni di ricorrenza:

$$d_{ij}^{(K,p,q)} = \begin{cases} \min\{e_0, e_1\} & \text{se } p=q=T \\ \min\{e_0, e_2\} & \text{se } p=T \wedge q=F \\ \min\{e_0, e_3\} & \text{se } p=F \wedge q=T \\ \min\{e_0, e_4\} & \text{se } p=q=F \end{cases}$$

Soluzione: $\forall (i,j) \in V^2$ $d_{ij}^{(n,T,T)}$