Technical Communication for Computer Scientists

Summer 2013

Final Report:
The Movie-Chain-Runner Project

June 26, 2013

Submitted to
Thomas M. Keating
Assistant Teaching Professor
Computer Science Department
Carnegie Mellon University

Prepared and Submitted by

Shashank Singh
sss1@andrew.cmu.edu

Eugene Scanlon
escanlon@cmu.edu

Abstract

We attempted to solve the Movie-Chain-Runner Problem, a computational problem equivalent to the well-known Longest Path Problem. We designed and used Python to implement several algorithms for the problem, and include our best solution. In the end, we accomplished our original goal of constructing a movie chain of at least 300 titles. We also performed some analyses of both the problem and our algorithms, and use them here to discuss approaches to the problem. Finally, we overview and reflect on our group's approach and success.

Contents

1 Introduction

We first introduce the Movie-Chain-Runner Problem, our solution, and our success.

1.1 The Movie-Chain-Runner Problem

The Movie-Chain-Runner Problem is to find the longest chain of overlapping titles in a list of movie titles, where two titles are said to overlap if some suffix of the first movie is identical to some prefix of the second movie. For example, in the list

- Day of the Dead
- Live and Let Die
- Dead Poets' Society
- Die Another Day
- The Last Samurai

the longest chain consists of 4 titles:

Live and Let Die Another Day of the Dead Poets' Society

Our project is to approximately solve the Movie-Chain-Runner Problem for a given list of 6561 movie titles. By appropriately representing the movie list as a graph, we found that the Movie-Chain-Runner Problem is equivalent to the Longest Path Problem (finding the longest simple path in a directed graph). The Longest Path Problem is well known to be NP-Hard, meaning that no efficient algorithm exists to find longest paths in large graphs, including our movie graph. A review of the literature reveals that good approximate longest paths also cannot be found efficiently in large graphs.[1] Thus, our project is to study the movie title graph and innovate ways to find long paths in the graph.

1.2 Solution

We constructed our 301 title chain in four major steps. Our first step was to reconstruct the list of movie titles as a directed graph, thus reducing the problem to the well-known Longest Path Problem. The second step was to implement a simple brute-force algorithm. The final two steps involved augmenting this brute-force algorithm with two new procedures: "reversal" and "replacement." The algorithms are described in detail under Section 2.3. on page 2.

1.3 Did We Succeed?

Our original goal was to surpass 300 titles. Despite some early setbacks leading us to lower our goal to 285 titles, we reached our original goal of 300 titles with 301 titles (see Appendix B for the full movie chain). We did not achieve the optional goal of 328 titles needed to set a new record.

[1] Björklund, Husfeldt, Khanna, "Approximating longest directed paths and cycles," ICALP, 2004. (accessed June 26, 2013 a `http://citeseerx.ist.psu.edu/viewdoc/similar?doi=10.1.1.105.262`).

2 Approach

2.1 Programming Languages

Our team considered programming our Python, C, or MATLAB. We settled on Python because our entire team was familiar with Python, whereas only three team members had used C, and only two had used MATLAB. Although C and MATLAB are faster languages, they are harder to use, and it would have taken too much time for team members to learn either. Jimmy also wrote Bash scripts to run our algorithms in parallel on cluster machines.

2.2 Algorithms

We present several algorithms (both successful and unsuccessful) that we considered.

Algorithms that worked:

Our first attempt at a solution was a brute-force algorithm which greedily followed every path in a depth-first search pattern, and retained the longest path found. We ran this algorithm until it ceased to make appreciable progress at a chain of 243 titles.

Our next step, "reversal," was to reverse all edges in the movie graph and use the brute force algorithm to extend the beginning of the 243 title chain. We felt this approach would be useful because, upon running our original algorithm multiple times, we found that the beginning of the chain changed significantly, but end of the chain was always the same, suggesting that the end of the chain was already nearly optimal. Again, we ran the algorithm until it ceased to make appreciable progress, this time with a chain of 277 titles.

Our final step, "insertion," checked if there was a longer subchain between two titles in our 277 title chain. Simply attempting to "squeeze" chains in between titles in our chain only increased our chain length to 278. However, when we allowed the insertion algorithm to replace entire subchains ("replacement"), we were able to construct our longest chain of 301 titles before the algorithm ceased to make much progress.

Algorithms that didn't work:

We tried an algorithm that computes acyclic subgraphs of our graph and then runs a polynomial time Longest Path algorithm known for acyclic graphs. However, the number of cycles proved too large to generate all the subgraphs, and so we abandoned this effort.

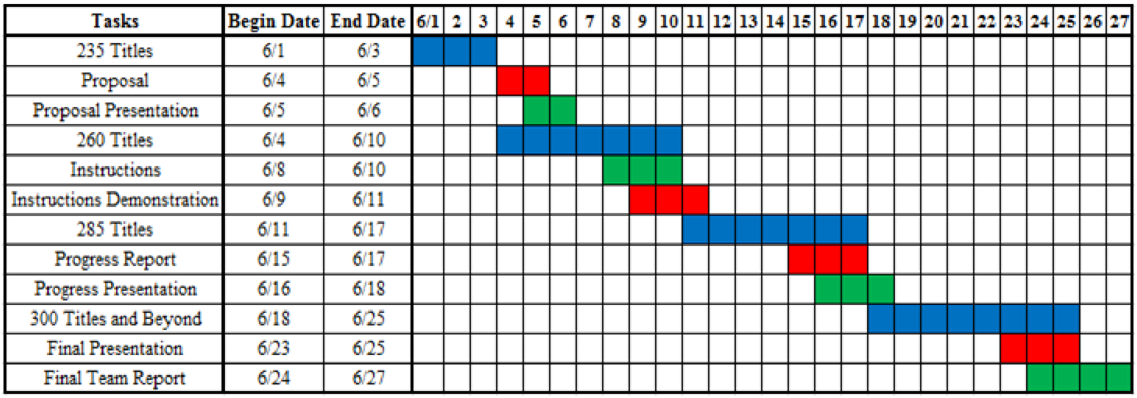
We also considered some more elaborate algorithms we found in the literature: a color-coding algorithm proposed by Alon et al.[2] and the genetic algorithms studied by Portugal et al.[3] However, due to the complexity of implementing these algorithms and the success of our simpler algorithms, we opted not to try them.

[2] Noga Alon, Raphael Yuster, Uri Zwick, Color-coding, Journal of the ACM (JACM), v.42 n.4, p.844-856, July 1995. (accessed June 16, 2013 at `http://dl.acm.org/citation.cfm?id=210337`).

[3] D. Portugal, C. H. Antunes, R. Rocha, "A Study of Genetic Algorithms for Approximating the Longest Path in Generic Graphs," Proc. of the IEEE SMC, pp. 2539-2544, 2010. (accessed June 16, 2013 at `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5641920& navigation=1`).

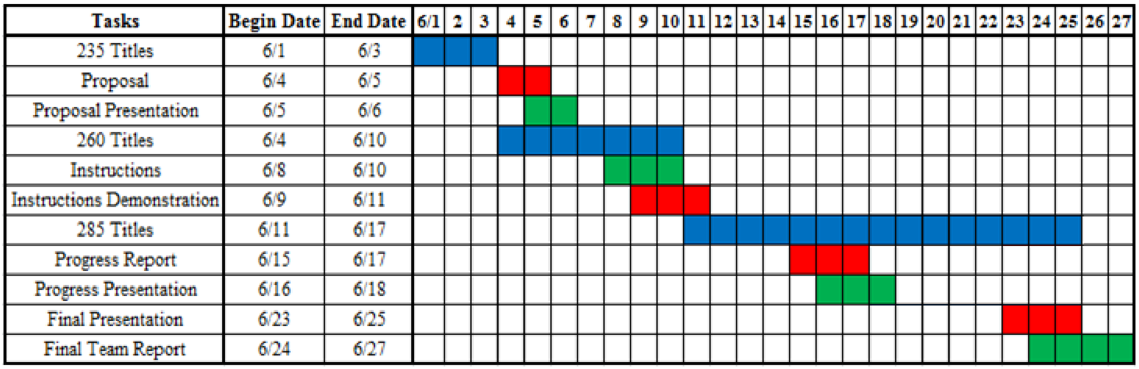
2.3 Schedule

Figure 1 below shows the original Gantt chart presented in our project proposal. Figure 2 below shows the revised Gantt chart we presented in our progress report. The only changes were that the 300 title goal originally set for June 25 was eliminated and the 285 title goal originally set for June 17 was extended to June 25. In practice, the only deviation from our original schedule was achieving our 285 title goal one day late (June 18), but we caught up by reaching our 300 title goal the next day (June 19).



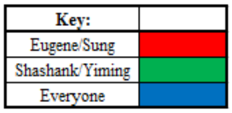
Tasks	Begin Date	End Date	6/1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
235 Titles	6/1	6/3	🟦	🟦	🟦																								
Proposal	6/4	6/5				🟥	🟥																						
Proposal Presentation	6/5	6/6					🟩	🟩																					
260 Titles	6/4	6/10				🟦	🟦	🟦	🟦	🟦	🟦	🟦																	
Instructions	6/8	6/10								🟩	🟩	🟩																	
Instructions Demonstration	6/9	6/11									🟥	🟥	🟥																
285 Titles	6/11	6/17											🟦	🟦	🟦	🟦	🟦	🟦	🟦										
Progress Report	6/15	6/17															🟥	🟥	🟥										
Progress Presentation	6/16	6/18																🟩	🟩	🟩									
300 Titles and Beyond	6/18	6/25																		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦		
Final Presentation	6/23	6/25																							🟥	🟥	🟥		
Final Team Report	6/24	6/27																								🟩	🟩	🟩	🟩

Figure 1: Our original Gantt chart.



Tasks	Begin Date	End Date	6/1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
235 Titles	6/1	6/3	🟦	🟦	🟦																								
Proposal	6/4	6/5				🟥	🟥																						
Proposal Presentation	6/5	6/6					🟩	🟩																					
260 Titles	6/4	6/10				🟦	🟦	🟦	🟦	🟦	🟦	🟦																	
Instructions	6/8	6/10								🟩	🟩	🟩																	
Instructions Demonstration	6/9	6/11									🟥	🟥	🟥																
285 Titles	6/11	6/17											🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦		
Progress Report	6/15	6/17															🟥	🟥	🟥										
Progress Presentation	6/16	6/18																🟩	🟩	🟩									
Final Presentation	6/23	6/25																							🟥	🟥	🟥		
Final Team Report	6/24	6/27																								🟩	🟩	🟩	🟩

Figure 2: Our revised Gantt chart.



Key:	
Eugene/Sung	🟥
Shashank/Yiming	🟩
Everyone	🟦

3 Results

We succeeded in finding a movie chain of 301 titles, fulfilling our scheduled goal of having a 300+ title chain by June 27.

Algorithm	Longest Chain
Brute-Force	243
Brute-Force w/ reversal	277
Brute-Force w/ reversal and insertion	278
Brute-Force w/ reversal and replacement	301

Table 1: Longest chain found by each algorithm

4 Discussion

Here we present the project evaluation promised in our proposal, followed by a discussion of lessons learned from the project and the resulting recommendations for future groups.

4.1 Evaluation

As suggested in our proposal, we evaluated the success of our project using three methods.

Method 1: Longest Chain Length

Our project was a success in that the number of titles in our longest chain exceeded 300. Although we failed to establish a new record, this secondary goal was optional and not expected.

Method 2: Algorithm Comparison

Our second method of evaluation was to compare different algorithms to decide on a "best" algorithm for the problem. Although we did not test a diverse selection of algorithms, we did try several variants on the basic brute force algorithm (as explained on page 2, under Section 2.3. Table 1 above presents the longest chain found by each variant. It is clear from the results that both reversal and replacement are important augmentations of the basic brute-force algorithm.

Method 3: Time Prediction

In order to guage the difficulty of the problem at hand, we also attempted to predict the time needed to solve the problem by exploring all possible paths in the movie graph. To do this, we first measured the time taken to solve the problem on many random subgraphs of the movie graph and used this data to fit an exponential relation between the size of the graph and the time taken for the computation. We then extrapolated this relation to conclude that performing the computation on the entire graph would take about 1.03 trillion years, approximately 100 times the lifespan of the sun. Appendix A gives a detailed discussion of the experiment and its results and validity.

4.2 Lessons Learned

We learned several things about programming for large computational problems:

- Using primarily brute-force algorithms, we came quite close to (likely within 13% of) the optimal solution. Thus, although far from optimal in theory, brute-force algorithms can produce decent solutions for some NP-hard problems in practice.

- For algorithms with large asymptotic runtimes, constant factors are important - a 50% runtime reduction is very significant when the algorithm can take a day or more to run. This fact is really understated in CS classes.

- With a high level language (like Python) it is crucial to consider all available data structures and their implementations. We noticed too late that using a set rather than a list for part of our code might have sped up our program by a factor of 100.

4.3 Recomendations

Based on our experience, we strongly recommend attempting to implement simple algorithms first when when tackling an NP-hard problem. While much of the literature is focused on optimizing asymptotic performance, in practice, a well implemented brute-force algorithm may perform nearly optimally with the time and hardware available, while being much simpler to implement correctly. Consequently, we also recommend trying to fully optimize code for such an implementation fairly early, since much time can be wasted in running the original program when one will later run a significantly optimized version.

Our team's work ethic and organization were crucial to our success. We recommend having frequent meetings (at least twice weekly) in order to coordinate work and ensure team members remain enganged.

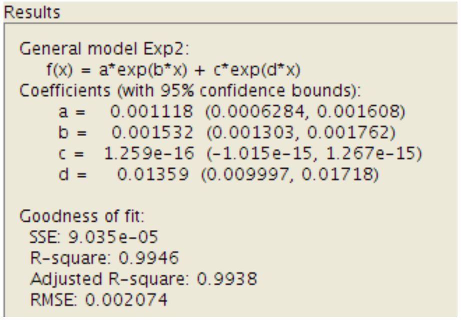
5 Sources

- Gantt Chart created using software from the Gantt Project (accessed June 4, 2013)
  - `http://www.ganttproject.biz/`
- Git repository hosted on GitHub (accessed June 4, 2013)
  - `https://github.com/`
- Python Programming Language (accessed June 4, 2013)
  - `http://www.python.org/`

Appendices

A Time Prediction

To guage the difficulty of the problem at hand, we predicted the time needed to solve the problem by exploring all possible paths in the movie graph. For each $k \in \{100, 200, 300, \ldots, 2500\}$, we first generated 100 random subgraphs, each induced by k titles chosen uniformly at random. We then measured the runtime on each subgraph, and averaged over samples to estimate the runtime for k titles. We fitted an exponential relation between the size of the graph and the runtime (the parameters of the exponential are included in table 3). Evaluating this exponential function at $x = 6561$ gives an estimate of **1.03 trillion years** needed to compute the solution for the entire 6561 title movie graph.



```
Results

General model Exp2:
    f(x) = a*exp(b*x) + c*exp(d*x)
Coefficients (with 95% confidence bounds):
      a =    0.001118  (0.0006284, 0.001608)
      b =    0.001532  (0.001303, 0.001762)
      c =   1.259e-16  (-1.015e-15, 1.267e-15)
      d =     0.01359  (0.009997, 0.01718)

Goodness of fit:
  SSE: 9.035e-05
  R-square: 0.9946
  Adjusted R-square: 0.9938
  RMSE: 0.002074
```

Figure 3: Exponential function from graph size to runtime

Figure 4 shows the fitted plot of how the runtime scales with the size of movie graph.

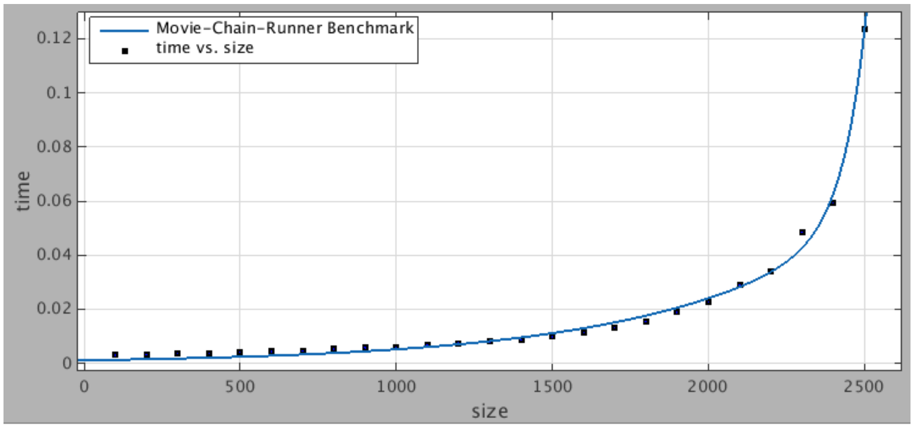


Figure 4: Plot of runtime (in seconds) over graph size (in titles).

The high R^2 value suggests that our data fit the exponential relation quite well. Since we measured runtime rather than operations performed, the results depend on our programming language and computer. The tests were run sequentially in Python on a single GHC cluster machine. However, a powerful computer running C would reduce the runtime by at most a factor of 10,000, and so the computation would still take millions of years.

B Our Longest Movie Chain

Here, we include our current longest movie chain, with 301 titles and 961 words:

```
THE GOSPEL
THE GOSPEL OF JOHN
JOHN Q
Q AND A
A PERFECT MURDER
MURDER AND MURDER
MURDER IN THE FIRST
FIRST BLOOD
BLOOD DIAMOND
DIAMOND MEN
MEN WITH GUNS
GUNS OF THE MAGNIFICENT SEVEN
SEVEN YEARS IN TIBET
TIBET CRY OF THE SNOW LION
LION OF THE DESERT
DESERT BLUE
BLUE CAR
CAR 54 WHERE ARE YOU
YOU CANT TAKE IT WITH YOU
YOU LIGHT UP MY LIFE
LIFE WITH FATHER
FATHER OF THE BRIDE
BRIDE OF THE MONSTER
MONSTER HOUSE
HOUSE OF DRACULA
DRACULA DEAD AND LOVING IT
IT TAKES TWO
TWO OR THREE THINGS I KNOW ABOUT HER
HER MAJESTY MRS BROWN
BROWN SUGAR
SUGAR AND SPICE
SPICE WORLD
WORLD TRADE CENTER
CENTER STAGE
STAGE FRIGHT
FRIGHT NIGHT
NIGHT FALLS ON MANHATTAN
MANHATTAN MURDER MYSTERY
MYSTERY ALASKA
ALASKA SPIRIT OF THE WILD
THE WILD ONE
ONE NIGHT STAND
STAND IN
IN OLD CALIFORNIA
CALIFORNIA SPLIT
SPLIT SECOND
SECOND BEST
BEST OF THE BEST
THE BEST OF EVERYTHING
EVERYTHING RELATIVE
RELATIVE FEAR
FEAR STRIKES OUT
OUT OF THE PAST
PAST MIDNIGHT
MIDNIGHT RUN
RUN SILENT RUN DEEP
DEEP BLUE
DEEP BLUE SEA
SEA OF LOVE
LOVE HAPPY
HAPPY BIRTHDAY TO ME
ME MYSELF I
I SPY
SPY HARD
HARD TIMES
TIMES SQUARE
SQUARE DANCE
DANCE WITH A STRANGER
STRANGER IN THE HOUSE
THE HOUSE OF THE DEAD
DEAD BANG
BANG BANG YOURE DEAD
DEAD END
END OF DAYS
DAYS OF HEAVEN
HEAVEN CAN WAIT
WAIT UNTIL DARK
DARK CITY
CITY OF JOY
JOY RIDE
RIDE THE HIGH COUNTRY
COUNTRY LIFE
LIFE IS BEAUTIFUL
BEAUTIFUL GIRLS
GIRLS GIRLS GIRLS
GIRLS JUST WANT TO HAVE FUN
FUN AND FANCY FREE
FREE WILLY
FREE WILLY 2 THE ADVENTURE HOME
HOME ALONE
ALONE IN THE DARK
DARK STAR
STAR WARS EPISODE V THE EMPIRE STRIKES BACK
BACK TO THE BEACH
BEACH PARTY
PARTY GIRL
GIRL IN THE CADILLAC
CADILLAC MAN
MAN OF THE HOUSE
HOUSE OF FRANKENSTEIN
FRANKENSTEIN AND THE MONSTER FROM HELL
HELL NIGHT
```

NIGHT AND DAY
DAY FOR NIGHT
NIGHT AND THE CITY
CITY OF ANGELS
ANGELS WITH DIRTY FACES
FACES OF DEATH
DEATH SHIP
SHIP OF FOOLS
FOOLS RUSH IN
IN COLD BLOOD
BLOOD SIMPLE
SIMPLE MEN
MEN IN BLACK
BLACK AND WHITE
WHITE LIGHTNING
LIGHTNING IN A BOTTLE
BOTTLE ROCKET
ROCKET MAN
MAN ON FIRE
FIRE IN THE SKY
SKY HIGH
HIGH CRIMES
CRIMES OF PASSION
PASSION IN THE DESERT
DESERT HEARTS
HEARTS OF DARKNESS A FILMMAKERS APOCALYPSE
APOCALYPSE NOW
NOW YOU SEE HIM NOW YOU DONT
DONT BOTHER TO KNOCK
KNOCK OFF
OFF THE BLACK
THE BLACK ANGEL
ANGEL HEART
HEART CONDITION
CONDITION RED
RED DAWN
DAWN OF THE DEAD
DEAD HEAT
HEAT AND DUST
DUST TO GLORY
GLORY ROAD
ROAD HOUSE
HOUSE PARTY
PARTY MONSTER
MONSTER IN A BOX
BOX OF MOON LIGHT
LIGHT IT UP
UP CLOSE AND PERSONAL
PERSONAL BEST
BEST MEN
MEN DONT LEAVE
LEAVE HER TO HEAVEN
HEAVEN AND EARTH
EARTH GIRLS ARE EASY
EASY MONEY
MONEY FOR NOTHING
NOTHING BUT TROUBLE
TROUBLE IN PARADISE
PARADISE ROAD
ROAD GAMES
GAMES PEOPLE PLAY NEW YORK
NEW YORK NEW YORK
NEW YORK COP
COP LAND
LAND OF THE DEAD
DEAD MAN
DEAD MAN ON CAMPUS
CAMPUS MAN
MAN TROUBLE
TROUBLE EVERY DAY
DAY OF THE DEAD
DEAD OF NIGHT
NIGHT MOTHER
MOTHER JUGS AND SPEED
SPEED 2 CRUISE CONTROL
CONTROL ROOM
ROOM AT THE TOP
TOP GUN
GUN CRAZY
CRAZY AS HELL
HELL UP IN HARLEM
HARLEM RIVER DRIVE
DRIVE ME CRAZY
CRAZY PEOPLE
PEOPLE I KNOW
I KNOW WHAT YOU DID LAST SUMMER
SUMMER CATCH
CATCH A FIRE
FIRE ON THE MOUNTAIN
THE MOUNTAIN MEN
MEN CRY BULLETS
BULLETS OVER BROADWAY
BROADWAY DANNY ROSE
ROSE RED
RED EYE
EYE FOR AN EYE
AN EYE FOR AN EYE
EYE OF GOD
GOD TOLD ME TO
TO DIE FOR
FOR YOUR EYES ONLY
ONLY THE STRONG
ONLY THE STRONG SURVIVE A CELEBRATION OF SOUL
SOUL FOOD
FOOD OF LOVE
LOVE IS THE DEVIL
THE DEVIL RIDES OUT
OUT COLD
COLD FEVER
FEVER PITCH

```
PITCH BLACK
BLACK HAWK DOWN
DOWN WITH LOVE
LOVE LIFE
LIFE OR SOMETHING LIKE IT
IT HAPPENED AT THE WORLDS FAIR
FAIR GAME
GAME OF DEATH
DEATH WISH V THE FACE OF DEATH
DEATH WISH
WISH UPON A STAR
A STAR IS BORN
BORN AMERICAN
AMERICAN HISTORY X
X THE MAN WITH THE X RAY EYES
EYES OF AN ANGEL
ANGEL BABY
BABY SECRET OF THE LOST LEGEND
LEGEND OF THE LOST
THE LOST BOYS
BOYS AND GIRLS
GIRLS WILL BE GIRLS
GIRLS OF SUMMER
SUMMER LOVERS
LOVERS AND OTHER STRANGERS
STRANGERS WHEN WE MEET
MEET JOE BLACK
BLACK LIKE ME
ME WITHOUT YOU
YOU ONLY LIVE ONCE
ONCE AROUND
AROUND THE BEND
BEND OF THE RIVER
THE RIVER WILD
WILD THINGS
THINGS TO COME
COME AND GET IT
IT HAPPENED ONE NIGHT
ONE NIGHT WITH THE KING
THE KING AND I
I WANT TO LIVE
LIVE AND LET DIE
DIE MOMMIE DIE
DIE MONSTER DIE
DIE HARD
HARD EIGHT
EIGHT AND A HALF WOMEN
WOMEN IN LOVE
IN LOVE AND WAR
WAR OF THE WORLDS
THE WORLDS FASTEST INDIAN
INDIAN SUMMER
SUMMER SCHOOL
SCHOOL OF ROCK
ROCK STAR
STAR TREK THE MOTION PICTURE
PICTURE BRIDE
BRIDE OF THE WIND
THE WIND AND THE LION
THE LION KING
KING OF THE JUNGLE
JUNGLE 2 JUNGLE
JUNGLE BOOK
BOOK OF LOVE
LOVE WALKED IN
IN GODS HANDS
HANDS ON A HARD BODY
BODY DOUBLE
DOUBLE TEAM
TEAM AMERICA WORLD POLICE
POLICE ACADEMY
POLICE ACADEMY 3 BACK IN TRAINING
TRAINING DAY
DAY OF THE WOMAN
THE WOMAN IN RED
RED RIVER
RIVER OF NO RETURN
RETURN TO HORROR HIGH
HIGH SCHOOL HIGH
HIGH SPIRITS
SPIRITS OF THE DEAD
DEAD MAN WALKING
WALKING AND TALKING
TALKING ABOUT SEX
SEX AND THE OTHER MAN
MAN OF THE YEAR
YEAR OF THE DRAGON
DRAGON SEED
SEED OF CHUCKY
```